AD-A035 847 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 9/2
NAVY LOGISTICS INFORMATION SHARING (NAVLIS) PROJECT.(U)
JUN 76

UNCLASSIFIED DTNSRDC-76-0120 NL

1 OF 5
AD
A035847

Report 76-0120

NAVY LOGISTICS INFORMATION SHARING (NAVLIS) PROJECT FINAL REPORT

ADA035847

DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Md. 20084

DEPARTMENT OF THE NAVY DTNSRDC

4

SC

NAVY LOGISTICS INFORMATION SHARING (NAVLIS) PROJECT FINAL REPORT

PREPARED BY
NAVLIS PROJECT OFFICE

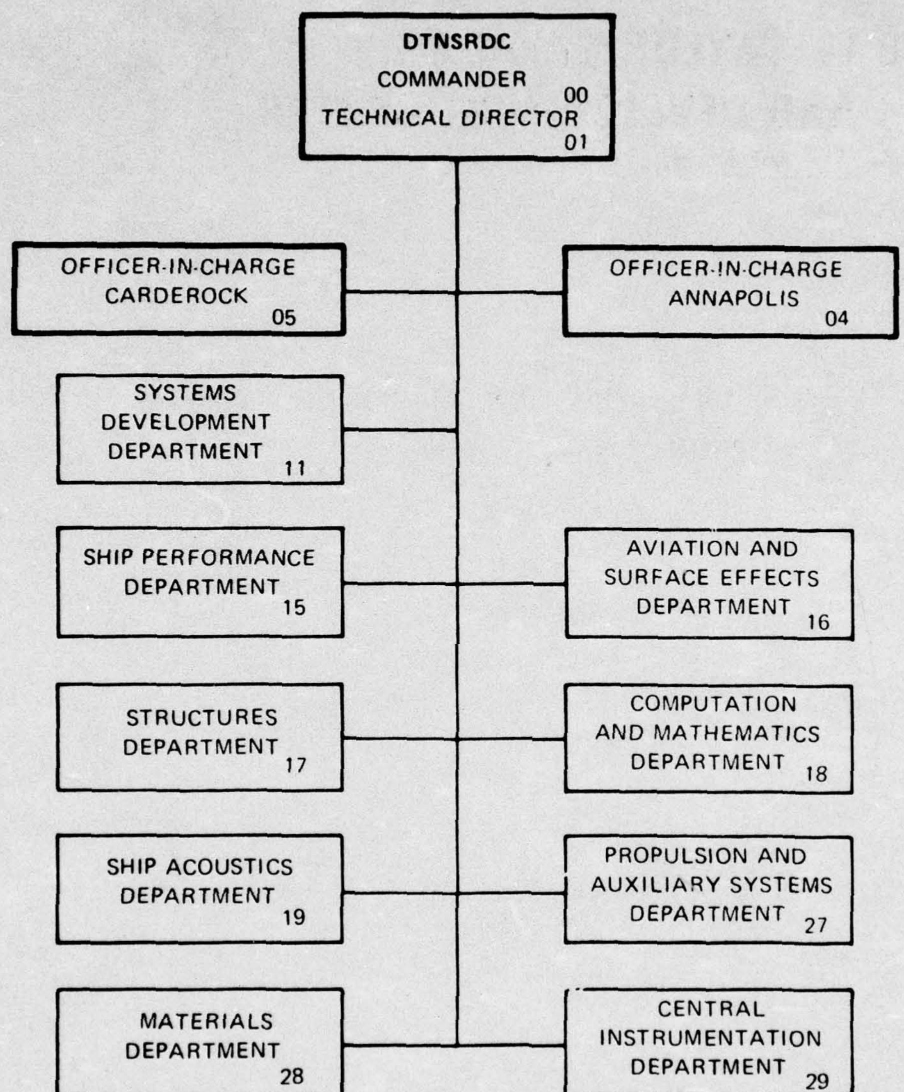
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

COMPUTATION AND MATHEMATICS DEPARTMENT
RESEARCH AND DEVELOPMENT REPORT

June 1976

Report 76-0120

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



GPO 9[illegible]-678

NDW-DTNSRDC 3960/43b (Rev. 11-75)
GPO 928-108

DTNSRDC ISSUES THREE TYPES OF REPORTS

(1) DTNSRDC REPORTS, A FORMAL SERIES PUBLISHING INFORMATION OF PERMANENT TECHNICAL VALUE, DESIGNATED BY A SERIAL REPORT NUMBER.

(2) DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, RECORDING INFORMATION OF A PRELIMINARY OR TEMPORARY NATURE, OR OF LIMITED INTEREST OR SIGNIFICANCE, CARRYING A DEPARTMENTAL ALPHANUMERIC IDENTIFICATION.

(3) TECHNICAL MEMORANDA, AN INFORMAL SERIES, USUALLY INTERNAL WORKING PAPERS OR DIRECT REPORTS TO SPONSORS, NUMBERED AS TM SERIES REPORTS; NOT FOR GENERAL DISTRIBUTION.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Report 76-0120	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER rept.
4. TITLE (and Subtitle) 6 NAVY LOGISTICS INFORMATION SHARING (NAVLIS) PROJECT, FINAL REPORT		5. TYPE OF REPORT & PERIOD COVERED 9 Final Jan 1973—June 1976 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) NAVLIS Project Office		8. CONTRACT OR GRANT NUMBER(s) 16
9. PERFORMING ORGANIZATION NAME AND ADDRESS David W. Taylor Naval Ship Research and Development Center Bethesda, Maryland 20084		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 63705N, T4112 17 T4112001, 1806-008
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 11 June 1976 13. NUMBER OF PAGES 467 12 467p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 14 DTNSRDC-76-Ø12Ø		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Networking; Data Base Management System (DBMS); ADS;
Information Retrieval; Management Information System; Logistics Support

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Navy Logistics Information sharing (NAVLIS) Project was a Navy RDT&E program in the advanced development (6.3) phase. The primary objective of the NAVLIS project was ~~the~~ application of computer networking and data base accessing technology to the current Navy automated logistics environment. The program was terminated in December 1975. This report describes the computer software developed during the construction of a pilot model and the

(Continued on reverse side)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

387682

(Block 20 continued~~)~~ fr p. 1473A)

capabilities of that software to extract data from distributed data bases on non-homogeneous computers at geographically dispersed sites. The major technical interests and contributions of the project ~~are~~ were:

- a distributed Data Directory concept;
- a "generalized" parameter-driven data base interface;
- a synonym resolution capability; and
- a secondary "hit" resolution capability.

The major recommendations forwarded in section 8.2 of th~~e~~is report are that:

- ~~that~~ NAVLIS technology be reviewed for its relevance to on-going university level projects;
- ~~that~~ NAVLIS technology be considered for its potential benefits to other logistics system developments, such as WWMICS and NAICOMMIS; and
- ~~that~~ consideration be given to utilizing the NAVLIS technology in networking the Navy Data Processing Service Centers.

Appendix A of this report contains the individual software program documentation and logic charts comprising the NAVLIS Executive Subsystem Documentation. (The NAVLIS Pilot Model Software consisted of two subsystems, (1) the Executive Subsystem, and (2) Telecommunications Subsystem (NTS)).

Appendix B of this report describes the NAVLIS effort for the Commander in Chief, Atlantic Fleet (CINCLANTFLT) which was on-going at project termination. The report, in the form of a Case Study, describes the application, which was to assist the Atlantic Fleet in its ship overhaul scheduling and funding and to provide an interactive data base as a management tool for the CINCLANTFLT staff.

1473B

TABLE OF CONTENTS

	Page
1. INTRODUCTION	1
1.1 BACKGROUND AND PURPOSE OF REPORT	2
1.2 PROJECT HISTORY AND FUNDING PROFILE	3
2. EXECUTIVE SUMMARY AND CONCLUSIONS	6
3. TECHNICAL PRODUCTS AND ACCOMPLISHMENTS	14
3.1 DISTRIBUTED DATA DIRECTORY	14
3.2 DATA BASE INTERFACE	17
3.3 SYNONYM RESOLUTION	17
3.4 SECONDARY HIT RESOLUTION	18
4. REQUIREMENTS ANALYSIS/FLEET APPLICATIONS	21
4.1 COMMANDER IN CHIEF, ATLANTIC FLEET (CINCLANTFLT) EFFORT	21
4.2 ATLANTIC FLEET DATA PROCESSING SERVICE CENTER (DPSCLANT) EFFORT	34
5. PILOT MODEL STATUS AT PROJECT TERMINATION	36
5.1 "EXECUTABLE" NAVLIS PILOT MODEL	36
5.2 "DEVELOPMENT" NAVLIS PILOT SYSTEM	39
6. NAVLIS MANAGEMENT CONSIDERATIONS	42
7. SUMMARY OF PROJECT EXPENDITURES BY DTNSRDC	44
7.1 FISCAL YEAR 1973 EXPENDITURES	44
7.2 FISCAL YEAR 1974 EXPENDITURES	44
7.3 FISCAL YEAR 1975 EXPENDITURES	45
7.4 FISCAL YEAR 1976 EXPENDITURES	46
8. TECHNICAL CONSIDERATIONS AND RECOMMENDATIONS	48
8.1 CONSIDERATIONS	48
8.1.1 GENERALIZED "ADS INTERFACE" CONSIDERATIONS	48
8.1.2 SYNONYM RESOLUTION CONSIDERATIONS	51
8.1.3 APPLICATIONS VS. ADHOC QUERIES CONSIDERATIONS	54
8.1.4 USER PARTICIPATION IN RETRIEVAL SELECTION CONSIDERATIONS	56

Page

8.1.5 DATA FILE UPDATE CONSIDERATIONS 60

8.1.6 DATA ACCESS RESTRICTION CONSIDERATIONS 63

8.2 RECOMMENDATIONS 67

ACKNOWLEDGMENTS 68

REFERENCE - NAVAL COMMAND SYSTEMS SUPPORT ACTIVITY (NAVCOSSACT) REPORT ENTITLED:

NAVLIS TELECOMMUNICATIONS SUBSYSTEM

NAVCOSSACT 06C001 P5-01

APPENDIX A - NAVLIS EXECUTIVE SUBSYSTEM SOFTWARE DOCUMENTATION 69

APPENDIX B - CINCLANTFLT NAVLIS APPLICATION CASE STUDY 295

LIST OF FIGURES

1 - Annual NAVLIS Funding History 5

2 - NAVLIS Pilot Model Configuration 7

3 - Shipyard (Depot) Level Maintenance 23

4 - Intermediate Maintenance Activity (IMA) Planning and Performance Accounting 28

5 - Organizational Maintenance and Performance Accounting 32

1. INTRODUCTION

The Navy Logistics Information Sharing (NAVLIS) project is a Navy RDT&E program in the advanced development (6.3) phase. The objective of the project was to develop and demonstrate a capability to provide Navy Managers with accurate, controlled, and timely logistics data and information by applying computer networking and data base sharing to existing and planned Naval Logistics Automated Data Processing (ADP) systems.

The technology presented in this report represents 2-1/2 years of extensive effort by a team of analysts and programmers at DTNSRDC and other Navy Installations. This report describes and discusses the NAVLIS project status at its termination by its sponsors in December 1975.

1.1 BACKGROUND AND PURPOSE OF REPORT

In January 1973, The Chief of Naval Material (CHNAVMAT) advised the Commander, Naval Electronics Laboratory Center (NELC) and the Commander, David W. Taylor Naval Ship Research and Development Center (DTNSRDC) that:

> "...to centralize the R&D support of Navy Logistics... it is desired to reassign the technical management of NAVLIS (Navy Logistic Information Sharing) from NELC to DTNSRDC."

A NAVLIS study team was established in the DTNSRDC Computation and Mathematics Department to accomplish the transfer of the project from NELC. The mission of the team was established as follows:

a. Review the requirements and objectives of NAVLIS as stated in source documents in relation to current logistics information requirements.

b. Evaluate the progress achieved by the NAVLIS Program to date.

c. Assess current efforts from the standpoint of their potential for resolving technological and operational problems.

d. Formulate a DTNSRDC management plan for NAVLIS.

The DTNSRDC evaluation listed the following general findings:

a. The NAVLIS development program had fallen short of the goals spelled out in the requirements documents. A system Initial Operational Capability (IOC) in FY75 was not attainable.

b. The NAVLIS development effort now differed markedly from that proposed in the Proposed Technical Approaches (PTA) and set forth as a requirement in the Advanced

Development Objective (ADO). Emphasis had shifted from system development to development of an information sharing capability.

c. Neither the reduction in program objective nor the shift in development emphasis had been formally documented.

d. The original requirement for improvements in Navy Logistics information management still existed.

This report, representing the final documentation of the NAVLIS project, identifies significant findings, and formulates recommendations and potential applications.

1.2 PROJECT HISTORY AND FUNDING PROFILE

The NAVLIS program began with the issuance of a Tentative Specific Operational Requirement (TSOR 41-12T) in December 1967, but was first funded as a 6.3 program in Fiscal Year 69. The NAVLIS Advanced Development Objectives (ADO 41-12X) was issued in December 1969. The OPNAV sponsor for NAVLIS is OP-04. The Principal Development Activity (PDA) is the Chief of Naval Material with specific responsibility assigned to MAT 034. The lead laboratory function was transferred from Naval Electronic Laboratory Center (NELC) to David W. Taylor Naval Ship Research and Development Center (DTNSRDC) in February 1973.

The NAVLIS TSOR, Proposed Technical Approaches (PTA 41-12T), ADO, and Project Management Plan (PMP) were issued prior to Calendar Year 1970. These documents were directed toward development and implementation of a large, Navy-wide, logistics information system. This system, as originally envisioned, had a dedicated management and operations staff, unique supporting hardware and communications, and unique and highly sophisticated software. Development funds were estimated at $19.7M in the PTA and $15M in the ADO to bring the system to Initial Operational Capability (IOC) in FY75.

Actual funding for NAVLIS is shown in Figure 1. Total funding through FY75 was \$3.07M or approximately 20% of the ADO plan. Funding in FY70 and FY72 was zero.

Working level responsibility as PDA has been assigned to four different groups within the Naval Material Command since 1968, and technical responsibility to two different Navy laboratories since 1970.

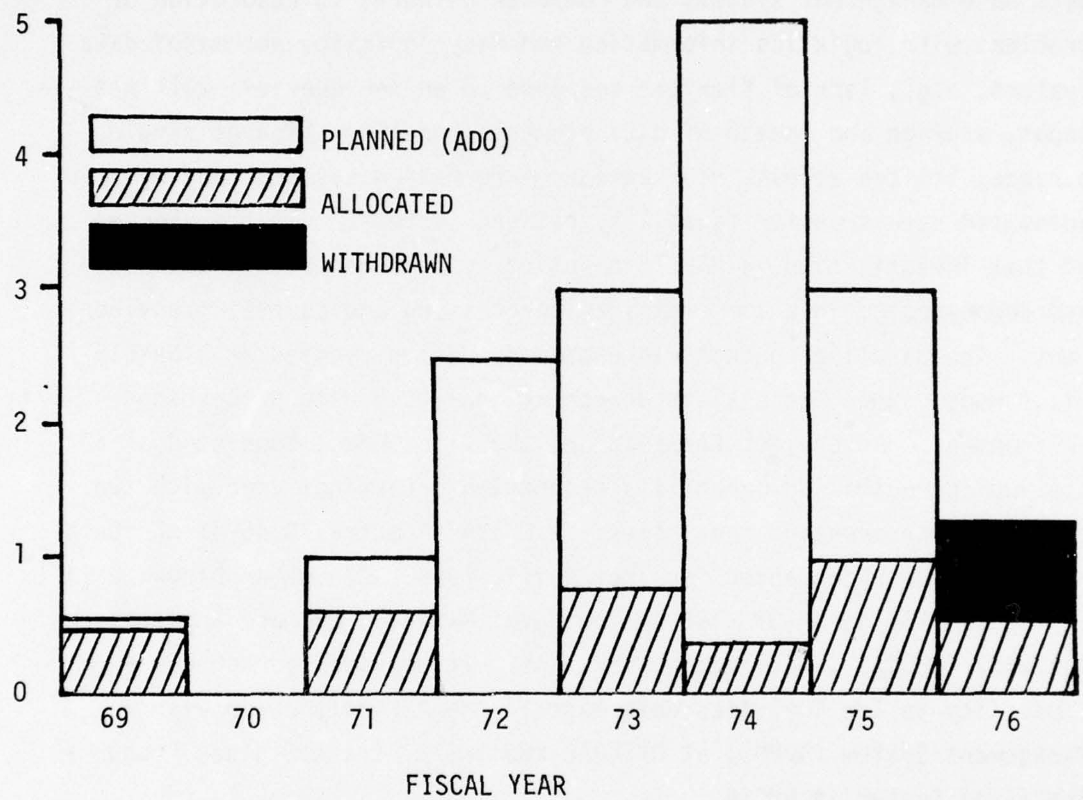


Figure 1 - Annual NAVLIS Funding History

2. EXECUTIVE SUMMARY AND CONCLUSIONS

The objectives of the NAVLIS project as stated in the Technical Development Plan (TDP 41-12X, 15 April 1974) were to "Investigate feasibility and desirability of applying developments in distributed data base management systems and computer networks to resolution of problems with logistics information and Navy logistics automated data systems, e.g., lack of flexible response to ad hoc queries; multiple input, storage and update of data elements; proliferation of single purpose, limited access, non-standard information systems; and limited automated data transfer capability between systems." In the process of that investigation, a NAVLIS baseline system concept was developed and demonstrated in a commercial teleprocessing and computing environment. The baseline concept was expanded and implemented as a NAVLIS Pilot Model (See Figure 2) in an operational Navy data processing environment. At project termination, the Pilot Model consisted of a dial-up communication capability connecting a terminal user with two remote data processing facilities: a UNIVAC Spectra 70/45 at the Data Processing Service Center for the Pacific (DPSCPAC) in San Diego, Calif. and an IBM 360/65 or 370/165 at the Naval Material Command Support Activity (NMCSA) in Arlington, Virginia. It provided ad hoc query capability to two logistics data bases: the Aircraft Power Plant Management System (APPMS) at DPSCPAC and the Aircraft Engines Accounting (AEA) System at NMCSA.

This section will address the areas of "feasibility and desirability" of a NAVLIS-type capability as well as questions of practicability, sponsorship, and implementation. The conclusions presented here are based on the experience gained during concept formulation and Pilot/Model development.

In discussing the feasibility of a NAVLIS-type system, technical feasibility must be separated from administrative or organizational feasibility. No new technological developments were required for the development and implementation of the Pilot Model and no requirement

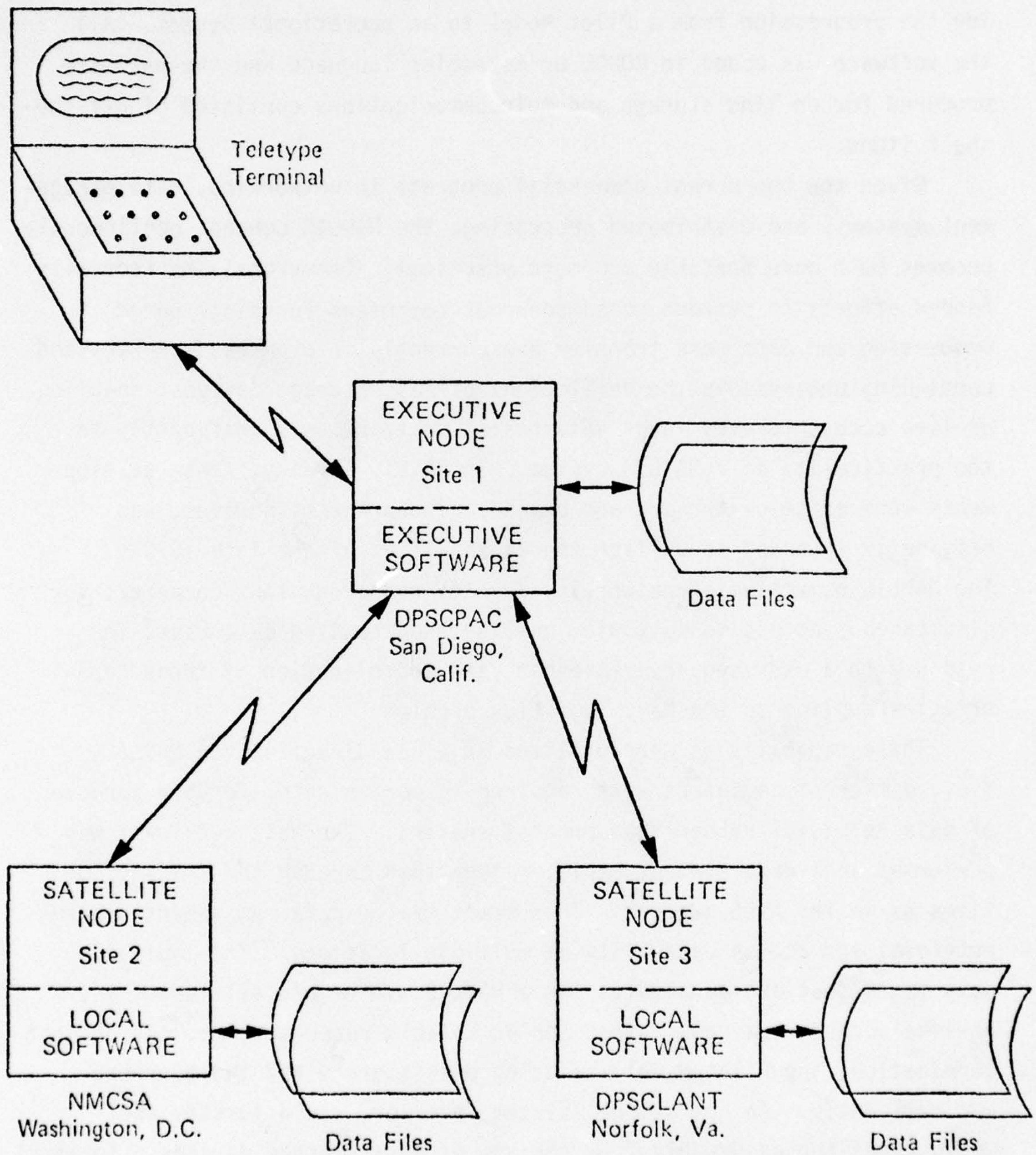


Figure 2 - NAVLIS Pilot Model Configuration

for any unavailable technological features was identified when considering the progression from a Pilot Model to an operational system. All the software was coded in COBOL or Assembler language and the hardware procured for on-line storage and telecommunications consisted of off-the-shelf items.

Given the concurrent commercial progress in networking, data management systems, and distributed processing, the NAVLIS concept continuously becomes both more feasible and more practical. Commercial and Federally funded efforts to network nonhomogeneous computers for distributed processing and data base transfer are currently in progress. Recent and continuing progress in the development of mass storage devices, enabling on-line access to very large data bases, contributes significantly to the practicality of a NAVLIS system. In NAVLIS infancy, these developments were state-of-the-art and beyond. The project, however, was originally intended to utilize the capabilities of the late 1970's. The NAVLIS objective of networking remote, nonhomogeneous computers for simultaneous access to multiple, on-line, nonstandard data bases in response to a user's query represents the consolidation of these capabilities applied to the Navy logistics problem.

These capabilities were utilized in a new direction for NAVLIS, i.e., different computers were required to communicate, for the purpose of data retrieval rather than process sharing. The data retrieval was performed on a data element basis, rather than through the transfer of files as in the ARPA network. This meant having data management system retrieval and access capability at multiple locations. The logistics data bases that are candidates for a NAVLIS system are all large, yet on-line access is a requirement for acceptable response time. At project termination, the Pilot Model was using disk storage and the overhead was noticeable. An operational system, however, could justify and would benefit considerably from the use of mass storage devices. In short, the NAVLIS technological requirements were very much in tune with commercial and government software and hardware developments, and the developments continue to establish the practicality and technical feasibility of the NAVLIS concept.

Feasibility in areas other than hardware and software is an entirely different matter. An especially difficult area concerns the individuals responsible for a data base and its contents, and their hesitancy to make their data available to other logistics managers. The tendency is to want to "sanitize" the data before it leaves the responsible manager's jurisdiction. The justification for this is that often raw data values can be misleading if not considered in conjunction with other values or conditions. These additional values or conditions may be represented by other contents in the data base, in which case the system can take some measures to see that all relevant data are retrieved or that the user is advised that the data retrieved may be incomplete or misleading (see Section 7.1.6). Little can be done, however, with data that normally gets manipulated by the local manager before release. When the values in the data base become meaningful or valid only when combined with nonautomated data, e.g., hardcopy reports or personal knowledge, the concern that erroneous conclusions may be drawn from exposure to the data base is a valid one. In this case, consideration should be given to automating the remaining values necessary to make the data base complete.

The "desirability" of the NAVLIS project became highly suspect at the time of, and was a major reason for, project termination. The project was the object of some criticism, quite a bit of it from competitive efforts. The NAVLIS system objectives and capabilities were in line with the needs of the Navy and its logistics managers, even those managers sponsoring competitive systems, but the NAVLIS method did not provide the tailored benefits found in the system designs of several of its critics.

The primary objective of the project was to provide logistics managers with access to data maintained in data bases not currently accessible to them. These data bases could be local or remote, would be on-line, and could be accessed by terminal to respond to ad hoc queries or to call for pre-programmed report generation. The need for this capability is emphasized by the several systems currently proposing to provide that capability to a considerably more restricted user community and for a smaller number of data bases. Often these efforts are directed toward a specific

logistics problem or problem set as opposed to a generalized capability.

As a typical example, at the time of NAVLIS project termination, the Naval Air Systems Command (AIR401) was seeking approval for a system * that would provide managers with access to data maintained in three existing and geographically separate data bases. At this point, the problems addressed and objectives of this effort and NAVLIS are the same. However, the non-NAVLIS approach pursued in this project requires a substantial upgrade in facilities, including new and larger computers. It also requires "centralization" of the data base under a new data management system, which in turn mandates data base restructuring and modification of existing application programs using those data bases. In anticipation of this new computing power, additional applications and enhancement to current applications were also a part of the system proposal. This approach, requiring significant new hardware procurement and major system development effort, is typical of many of today's logistics information system developments.

By contrast, under the NAVLIS approach to the same problem, the data bases to be accessed remain at their current locations, although they are placed on-line and inverted on selected data elements. This normally requires that additional on-line storage be provided. A telecommunications capability is provided to these sites to be accessed, if required. The NAVLIS software modules are installed on the current computers at each site. In the Pilot Model network, additional memory was provided at DPSCPAC to support the system software. Terminals are also required for the managers who query the system. Although some additional capability and capacity is normally a requirement for NAVLIS implementation, it is limited to that necessary to accomplish the primary objective, i.e., providing managers with on-line access to data bases currently inaccessible to them. The capabilities of data management

---

* This system is described in the Automated Data System (ADS) Development Plan for the Naval Aviation Logistics Data Analysis (NALDA) System of June 9, 1975 published by the Naval Air Systems Command (AIR 401).

system access are provided without relocation or restructuring of the data base. As a result, the organization and procedures currently established for maintaining that data base are not disrupted, and existing application programs that utilize that data base need not be modified or transferred to another computer (with the normally attendant conversion problems).

The NAVLIS approach does not require major facility upgrades. The system concept is aligned toward using existing resources to the greatest extent possible and toward the least possible impact on the operating environments involved. As such, the project often came in conflict with other efforts whose objective of providing better data access to logistics managers was closely linked with major hardware procurements.

The need for a NAVLIS-type system is perhaps more valid now than in 1968 at the inception of the project. Primary concern at that time centered around:

(a) the unavailability of current and reliable data to assist in managerial decision-making

(b) the maintenance of duplicate data at multiple locations as a result of an inability to share data

(c) the increased reporting burden imposed on the fleet to supply input to these redundant data bases

(d) the proliferation of problem-oriented data processing systems directed toward a restricted group of users as opposed to the development of a lesser number of generalized systems serving larger segments of the management community.

Each of the needs above is undoubtedly more of a problem today than it was a decade ago.

The continuing proliferation of problem specific systems that attempt to consolidate existing but separate data bases by combining them into larger data bases through "centralization" and "integration" is evidence that managers still need data that is being automated, but is not readily accessible. Some progress toward elimination of duplication in several data bases is accomplished through consolidation, but on a very limited scale. The amount of data that is maintained to support logistic management decisions is awesome. Individual systems for limited areas of logistics access large quantities of data, and the practical limits of consolidation are rapidly reached when dealing with data collections of this size. The potential data pool available through distributed data base access far exceeds that which can be accommodated through centralization, and avoids the problems associated with data restructuring that normally accompany centralizing and integrating.

The continued multiplication of application specific systems can be attributed to a reluctance to accept "generalized" systems. Although generalization as a design concept is much lauded, in practice it is often received with skepticism largely because of the tendency to compare the effectiveness and efficiency of generalized systems with alternative designs customized for specific applications. Such comparisons are invariably done on a one-to-one basis. The result is that "generalized" System A, for one million dollars, is determined to be somewhat less efficient than "customized" System X, also for one million dollars. The same conclusion is reached in other evaluations when System A is compared with customized Systems Y and Z, also for one million each. The independence of the evaluations could well produce three specialized systems at a cost of three million dollars. In fact, the generalized system may well have satisfied most of the requirements for each of the customized systems at a considerable savings. In this example, a half million dollars of specialized capability for each System X, Y, and Z could be added to the generalized system to

accommodate their specific requirements and a cost savings would still be realized.

Although the example given is hypothetical, the point is that evaluation of system design efficiency is performed in a manner disadvantageous to generalized systems. As NAVLIS was very much a generalized system, aimed at satisfying a broad spectrum of users, it was quite vulnerable to this type comparison. Explaining to a group of aircraft engine maintenance managers (who desire an information system for their specific requirement) that they should support the NAVLIS effort because, although it will support only 80% of their requirements, it can also be used by ship's maintenance personnel, is likely to get a chilly reception.

These factors made it difficult to generate user backing for the NAVLIS project. Original project sponsorship was at a level at which Navy-wide interests predominated and the generality of the system concept had appeal. However, prior to project termination, logistics managers, whose interests are necessarily more parochial, were called on for support and, as a result of the response received, the project was cancelled.

3. TECHNICAL PRODUCTS AND ACCOMPLISHMENTS

No technological or software "breakthroughs" were required to accomplish the NAVLIS objectives. The system objectives represented an expansion and integration of capabilities already available in special purpose programs or data management systems which were sufficient to meet those goals. Data element retrieval in response to an ad hoc query, for example, is an existing capability of many software systems. However, performing that same function on multiple data bases located at remote sites, simultaneously, is a significant expansion of that data retrieval capability. A list of these expansions or innovations peculiar to NAVLIS would include the distributed data directory concept, the data base interface, synonym resolution, and secondary hit resolution capability. Although these techniques are not new in themselves, their integration represents a new application of proven software approaches to solve the NAVLIS problem.

3.1 DISTRIBUTED DATA DIRECTORY

An original concern of the project design centered around the method to be used in responding to a user's query. It was clear that, in order to achieve acceptable response times, some form of random access index or inverted list, based on the values of selected data elements in the file to be accessed, would be required. The most efficient use of such lists would involve co-locating the inverted files with their associated data bases. This distribution of the inverted query lists throughout the network would insure that only one copy of each inverted file would be maintained. However, to avoid accessing all data bases for each query, it became very important that the network be able to identify as early as possible within the NAVLIS query processing cycle the specific data files which contained the data elements necessary to satisfy the conditions expressed by a user's query. To accomplish that objective, the Data Directory Concept, the definition of each data base's content (data elements,

etc.), was selected. The Data Directory, defining the contents of all data bases within the network, would be located at each site within the network and would be the primary factor in the implementation of an efficient, distributed query capability. This directory makes possible a partial query determination of the applicable data bases at the user's site. The final query resolution, with the use of efficient inverted files, would occur at the location of the data base. The application of these two principles, particularly the Data Directory, would minimize data transmissions, eliminate "no data" responses from remote sites, and identify at the earliest opportunity queries to which the system could not respond.

A File Content Directory was developed that identified the contents of the data bases in the system by data element type, e.g., aircraft model, engine type, hull number, federal stock number, etc., with the name and location of the data base. Using this directory, the File Selection Module (FSM) can analyze the user's query and determine which data bases in the system contain the types of data elements required to answer the query. For example, to survey the casualty reporting on 5" 54 gun mounts on attack aircraft carriers for 1975, the user might submit the following query:

```
IF SHIP-TYPE IS CVA
AND EQUIP-IDENT-CODE IS GB17
AND CASREP-DATE BE 1/1/75, 12/31/75?
PRINT SHIP-NAME, CAUSE-OF-INCIDENT.
```

("BE" in the query above, means "between or equal to")

Using the File Content Directory, the FSM can determine which data bases in the system have the types of data elements expressed in the query constraints (SHIP-TYPE, EQUIP-IDENT-CODE, CASREP-DATE) and at least one of the data element types requested for retrieval (SHIP-NAME, CAUSE-OF-INCIDENT). The data bases that satisfy these requirements are "candidates" for answering the query, and the query is sent to the sites at which those data bases reside. Data bases not satisfying the above requirements are eliminated from any further consideration.

Although the candidate data bases have the types of data elements needed to respond to the query, they may not have the values necessary to respond. A candidate data base may have the data element SHIP-TYPE, but perhaps none of the values for SHIP-TYPE in that data base will equal "CVA", in which case that data base cannot respond to the query. Determining whether the required values are present is a function of the Primary Hit Resolution Module using the inverted files which identify the values in that data base for each data element on which the data base is inverted. Consequently, some "no-data" or "no-hit" responses would still occur, but considerably fewer would occur as a result of the File Content Directory's preselection of data bases than would be the case if the query were transmitted to all sites regardless of the data content of their files.

The directory function would be distributed throughout the network. The Executive sites (those sites capable of accepting and analyzing a query) would each have a copy of the smaller, less volatile, and more maintainable component, the File Content Directory, and all system sites would have the inverted files that pertain to their data bases.

3.2 DATA BASE INTERFACE

One of the most challenging requirements of the NAVLIS system was that it retrieve data from existing data bases without restructuring or reformatting those data bases. The objective was to gain access to the file without causing extensive, or even moderate, changes in the application programs currently using those files. This meant that NAVLIS had to be equally capable of interfacing with data files designed for second generation batch processing and with those structured under sophisticated data management systems. In the early stages of the project, for both expediency and experience, a unique module was developed to achieve the interface with each data base. By project termination, however, a generalized data base interface had been defined and was in development (see Section 7.1.1 for discussion of the Generalized "ADS Interface"). The generalized program would have enabled a single module to access several files at a single site. It would also have minimized the effort required to include a new data base in the NAVLIS network.

3.3 SYNONYM RESOLUTION

The requirement for a synonym resolution capability sprang directly from the requirement to access multiple data bases without modification. There is little, if any, standardization across data bases, even when the area of interest, e.g., aircraft engine maintenance, is the same. In some cases the only apparent similarity is in the data element names, e.g., aircraft type, ship name. In one data base, "aircraft type" may include the series and model number, whereas in a second data base all three elements may be recorded as separate fields. A data element such as "ship name" presents additional problems in that there may be several permutations within the same data base for the name of a specific ship. Additionally, an item may be referred to by a code number in one file and by name in another. These differences in identification required resolution so that a user could retrieve data relevant to his query without entering multiple identification types for the item of interest.

Additional discussion of the Synonym Resolution Module may be found in Section 8.1.2.

The significance of the synonym resolution capability is not only that it provided a method of overcoming a lack of data element standardization between and within data bases, but that the terminal user has considerably more flexibility in phrasing and entering his query than before. The user can phrase the query in the nomenclature most familiar to him and be assured that it will be translated as appropriate at various other locations to retrieve the data requested. In both the Pilot Model and the San Diego demonstration, commonly accepted abbreviations and acronyms were included to facilitate querying, rather than trying to rectify conflicts across data bases. Acronyms such as "CINCPACFLT", "CPF", "COMNAVAIRPAC", "CNAP", AND "DPSCPAC" are many times easier to use in query construction than the expanded titles they represent.

Synonym resolution created some problems that were still to be resolved at project termination (see Section 8.1.2). The difficulties arose primarily when synonym resolution was combined with certain options available in the query language. Those problems, however, are surmountable and their difficulty is far outweighed by the benefits derived from synonym resolution.

3.4 SECONDARY HIT RESOLUTION

Random access was clearly a requirement for the NAVLIS system to provide a reasonable response time for on-line users. Since the structures of the data bases to be accessed were not to be modified, the addition of pointers or links, or the reorganization of the data base records into some logical relationship, were eliminated as methods of providing random access. The creation of inverted files as indices to the contents of the data bases, and the use of the Indexed Sequential Access Method, enabled random access without adding to or restructuring the data bases. The Primary Hit Resolution Module (PHRM) uses the inverted files to perform hit resolution. If, for example, the following

query is input:

```
IF ITEM-NAME IS ABC
AND STATUS-CODE IS A1
PRINT ITEM-NAME, DATE-REPAIRED.
```

PHRM accesses the inverted files and determines the record keys for those records that have the value "ABC" in the ITEM-NAME field. The record keys for words containing "A1" in the STATUS-CODE field are also determined. PHRM examines the two groups of keys and any record key found in both groups satisfies the conditions of the query. When all the data elements expressed in the query conditions or constraints (ITEM-NAME and STATUS-CODE in the example given) have been inverted, PHRM can identify the specific records to be accessed for data retrieval. Inversion, however, is not performed on all data elements in a data base. Inverting on too many data elements and on data elements with many unique values in a data base, such as quantities, can easily produce an inverted file that exceeds the size of the data base. To keep the size of the inverted files reasonable, only selected data elements are inverted. Those data elements selected should be those most likely to appear as constraints in the majority of queries against that data base.

The user of the system, however, should be able to retrieve data regardless of whether the data base has been inverted on all the constraints in the query. The fact that a query is entered with constraints that were not selected for inversion should not invalidate that query. The system objective was to be able to retrieve on non-inverted data elements. This capability alleviates both the requirement for total data base inversion and for the user to be aware of which data elements were inverted in any specific data base. In the example given, either ITEM-NAME or STATUS-CODE or both could be non-inverted data elements. If STATUS-CODE was not inverted, PHRM would access the inverted files to determine the record keys with the value "ABC" in the ITEM-NAME FIELD, These record keys would be passed to the ADS Interface Module along with an indicator from PHRM that only partial hit resolution had been performed. The ADS Interface Module would then access each record identified by PHRM

and test the contents of the STATUS-CODE field against the constraint value "A1". Those records that satisfied that test would satisfy the query constraints and the ADS Interface Module would retrieve the requested data elements. Although the example used here is extremely simple, the ADS Interface Module under development at project termination included the capability to resolve all the search criteria expressible in the query language.

The ADS Interface Module also included the capability to retrieve data when none of the query constraints were inverted. This involved reading the entire data base sequentially, since without inversion PHRM is unable to do any preliminary hit resolution. The response time for this type of search is prohibitive except when access to the data is a requirement and no other query formulation will produce the same results.

The secondary hit resolution capability was incorporated in the NAVLIS system so the user would not be restricted in his query input to those data elements that were inverted, and so that the number of data elements selected for inversion could be reduced while still providing acceptable response time. Response time does increase when secondary hit resolution is required, but the cost in time is usually worth the increased accessibility of data. An exception would be when the entire data base must be read to answer the query. The user would then have to determine his priorities among time, money and data.

4. REQUIREMENTS ANALYSIS/FLEET APPLICATIONS

At the time of NAVLIS project termination, CINCLANTFLT, Norfolk, Va. and the Atlantic Fleet Data Processing Service Center (DPSCLANT) were involved with on-going NAVLIS efforts. Both efforts are described in the following paragraphs.

4.1 CINCLANTFLT EFFORT

The CINCLANTFLT effort dealt with ship overhaul funding and scheduling in the Atlantic Fleet in support of the Maintenance and Material Readiness Division and is described in greater detail in Appendix B of this report. This application was designed to permit the ready retrieval of data to facilitate the production of the Monthly Report of Overhaul Expenditures. This report, which receives wide distribution at the NAVMAT, CNO, SYSCOM, and Fleet staffs, tracks the performance of the fleet overhaul program. The fleet overhaul program entails, over a seven-year period, the scheduling and budget programming for over 300 ships. The annual maintenance program is in excess of $300 million. As an extension of this application, the following efforts were being initiated at project termination:

(a) <u>Investigation of the Maintenance Planning and Performance Accounting System</u>

The purpose of the Maintenance Planning and Performance Accounting System is to assist the planning for allocation of resources for shipyard, intermediate level, and shipboard maintenance, and to account for their accomplishment. In the current application, maintenance resources are funds and personnel. Additionally, the system facilitates the preparation and submission of the several inputs required by the annual budget cycle, provides the basis for evaluating the effects of alternative funding schedules on fleet maintenance, and serves as a basis for evaluating the effectiveness of the fleet maintenance program.

The Maintenance Planning and Performance System is depicted in Figure 3 for the Depot or Shipyard level, in Figure 9 for the Intermediate Maintenance Activity level, and in Figure 5 for the Organizational or Shipboard level. The following section discusses the features of the system for accounting for each level of maintenance activity. Data sources and requirements are keyed, by number, to the appropriately numbered blocks on each figure and to the following paragraphs.

(b) Investigation of the CINCLANT Maintenance Planning and Performance Accounting System which encompasses the following areas:

1) Shipyard (Depot) Level Maintenance (Figure 3)

The point of entry to the Maintenance Planning and Performance Accounting System has been the Monthly Report of Overhaul Expenditures. This report shows how well the annual overhaul program is being executed. It lists, by ship, projected shipyard overhaul start and departure dates, projected fund availability, and the actual start and departure dates as well as the actual expenditures to date. Experience has demonstrated that approximately two man-weeks are required at each TYCOM and an additional two man-weeks are expended at the CINCLANTFLT level for the manual production of this report.

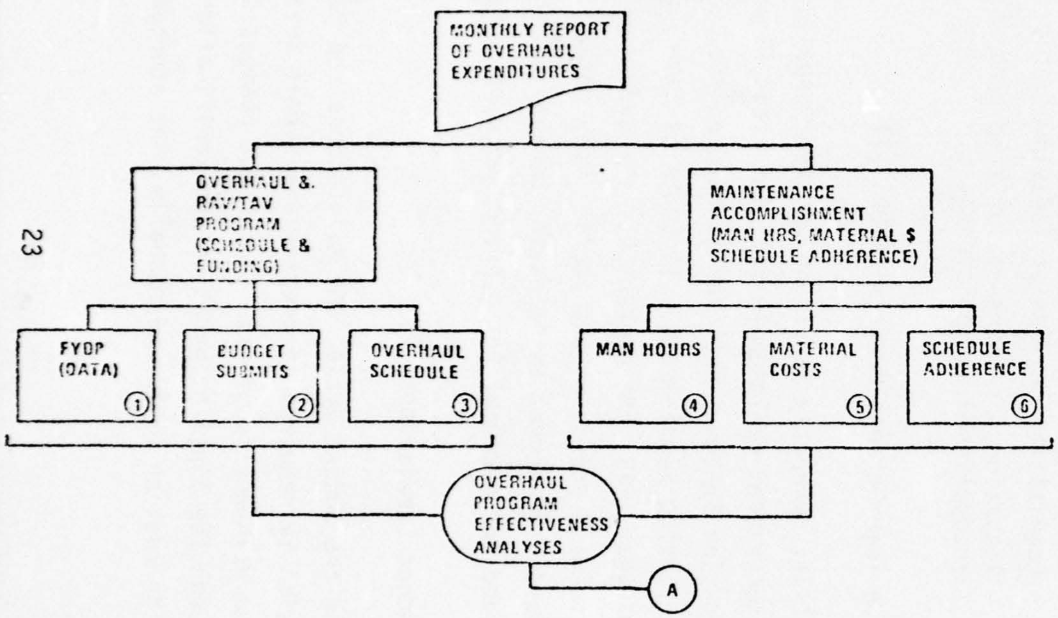
The platen dimensions of the presently available computer terminal preclude the on-line production of the final report. However, the data for report preparation can be assembled in less than thirty minutes.

2) Overhaul and Restricted Availability and Technical Availability Program (Schedule and Funding)

The production of the Monthly Report of Overhaul Expenditures requires data pertaining to both the planned maintenance program and the program accomplishment. The Overhaul and RAV/TAV Program

NAVLIS
CINCLANTFLT
MAINTENANCE & MATERIAL READINESS DIVISION
MAINTENANCE PLANNING & PERFORMANCE ACCOUNTING

SHIPYARD (DEPOT) LEVEL MAINTENANCE



NAVLIS INTERFACES

FUNCTIONAL FILE	PRESENT (PHASE I)	FULL NAVLIS (PHASE II)
1 FYDP	MANUAL	FMIS
2 BUDGET SUBMIT	MANUAL	MANUAL
3 OVERHAUL SCHED	MANUAL	SAMIS SECAS
4 MAN HOURS	DEPARTURE REPORT (MANUAL)	SHIPYARD MIS SAMIS
5 MATL COSTS	DEPARTURE REPORT (MANUAL)	SHIPYARD MIS SAMIS
6 SCHED ADHERENCE	SHIP/TYCOM MSG (MANUAL)	SHIP PROG REPORTS MSGS

Figure 3 - Shipyard (Depot) Level Maintenance

comprises three principal data files, the Five-Year Defense Program (FYDP), the series of budget submissions prepared by the Maintenance and Material Readiness Division, and the Schedule of Ship Overhauls. Data from these files can be used analytically to:

- Prepare further budget submissions
- Examine the effects of
  - Slippage of scheduled overhauls
  - Cancellation of scheduled overhauls
  - Reduction in ship maintenance program funding
  - Extension or compression of intervals between overhauls
  - Acquisition or loss of individual ships
  - Reduction, or increase, of time (and work) accomplished during overhauls.

3) Five-Year Defense Program (FYDP) (1)

This file, currently input and updated manually, will interface in later system refinements with the Fleet Financial Management Information System. The FYDP is used as the base line maintenance funding requirement. It contains data for the previous fiscal year (FY), the current FY, the budget year (BY), and five forward years, or out years. This departure from the standard FYDP format is necessary because a ship can be in a shipyard for overhaul for a period encompassing three FY's.

4) Budget Submissions (2)

During the budget cycle, the Maintenance and Material Readiness Division is required to prepare and submit several modifications to the maintenance program to reflect changes in guidance. Although the final preparation and verification of each submission must be performed manually, the storage, retrieval,

and comparative analysis of successive submissions is facilitated by the system. Data element descriptions contained in this file are similar to those in the FYDP file.

5) Overhaul Schedule (3)

The overhaul schedule reflects, by ship, the proposed start and departure dates for shipyard maintenance. Although this schedule is currently produced manually, it does interface with similar data residing in the Ship Alteration Management Information System (SAMIS), and to a lesser extent in the Ship Equipment Configuration System (SECAS). The implementation of a NAVLIS type network would permit automated cross update and retrieval of overhaul schedules. Although these three separate files are described individually, they are interrelated, in that changes to the funding program can, and do, create changes in the shipyard availability. Conversely, changes in shipyard availability can, and do, create changes in the funding program.

6) Maintenance Accomplishment

At the CINCLANTFLT level, maintenance accomplishment is measured in terms of material costs and man hours (or days) expended, and of course, in terms of adherence to the basic overhaul schedule.

7) Man Hours (4) and Material Costs (5)

Man hours (or man days) and material expenditures are currently reported in the Departure Report, prepared and submitted by the shipyard upon completion of an overhaul. Although the changes, in dollars, for labor are reflected in the report, analyses of program effectiveness and comparison of shipyard productivity are based on man hours or man days expended to accomplish scheduled overhauls.

Later phases of the application should interface with the Shipyard Management Information Systems, whenever they are implemented.

8) Schedule Adherence (6)

Messages received by the Maintenance and Material Readiness Division announce the arrival, progress, completion, and departure of ships at maintenance facilities. These messages are manually entered into a file and reflect the status, or progress, of the overhaul. Comparison of the data in this file with that contained in the overhaul schedule shows the adherence to, or departures from, the approved schedule, and serves as a flag for appropriate staff action.

9) Overhaul Program Effectiveness Analyses

Although there is currently no ad hoc NAVLIS capability for extracting data from a file, bringing the data into a work space, performing mathematical operations, and returning the results to the same or a new file, this capability is recognized as a requirement for any NAVLIS type system. Most commercial time sharing systems provide for such operations. Hence, the current system at CINCLANTFLT possesses no automated analytic capability. Data for analyses are extracted from the files and manually manipulated.

Program effectiveness analyses comprise:

- Analysis of labor and material expenditures for like work at different shipyards for similar ships.
- Analysis of labor and material expenditures at the same shipyard for both similar and different ships.
- Analysis of programmed expenditures, with actual charges.
- Analysis of expenditures for overhauls versus age of ships.
- Analysis of historical overhaul expenditures for like ships of the same age at the same and different shipyards.

(c) <u>An Investigation of the Intermediate Maintenance Activity (IMA) Planning and Performance Accounting - Figure 4</u>

Actual implementation of the IMA Module and the Organizational Maintenance Module of the Maintenance Planning and Performance System was not accomplished. Description of desired data, file content, and analytic operations were derived from discussions with the staff of the CINCLANTFLT Maintenance and Material Readiness Division.

The basic structure and concept of operation for the IMA and Organizational Modules is similar to that for the Shipyard (or Depot) Maintenance Modules. The schedule of work and associated plans for resource allocation are compared and contrasted with reports of actual maintenance performance and resource expenditures. In some cases, the present availability of data is limited. For example, the only data available on shipboard material consumption was in the supply system report of material expenditures. However, the systems described herein have been designed to start operations immediately using the presently available data.

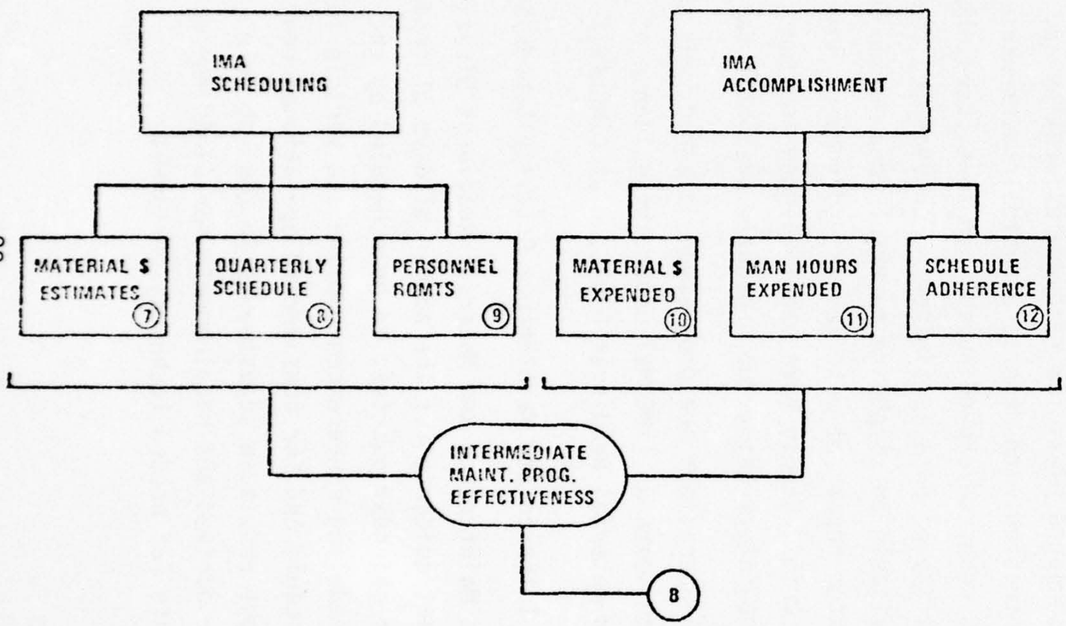
IMA operations are programmed, planned, and supervised by the Type Commands (TYCOMs) under the guidance of the Maintenance and material Readiness Division of CINCLANTFLT.

1) Intermediate Maintenance Activity Scheduling

The Maintenance and Material Readiness Division announces the Fleet guidance for the number of days of tender availability, per calendar quarter, to be scheduled by the TYCOM. Quarterly scheduling conferences provide the vehicle for firming up the schedule and for considering operational requirements which might result in adjustments in the schedule. From this schedule are derived the requirements for funds for material and requirements for needed technical personnel.

NAVLIS
CINCLANTFLT

INTERMEDIATE MAINTENANCE ACTIVITY
PLANNING AND PERFORMANCE ACCOUNTING



NAVLIS INTERFACES

FUNCTIONAL FILE	PRESENT (PHASE I)	FULL NAVLIS (PHASE II)
7 MATL ESTIMATES $	MANUAL BUDGET PROG	FMIS
8 QUARTERLY SCHED	TYCOM/CINCLANT MANUAL	MANUAL
9 PERSONNEL RQMTS	TENDER/TYCOM/ MANUAL	PERSONNEL MIS SECAS
10 MATL $ EXPENDED	IMMS MANUAL	IMMS
11 MAN HOURS EXPENDED	IMMS MANUAL	IMMS
12 SCHED ADHERENCE	SHIP/TYCOM IMMS (MAN.)	IMMS

Figure 4 - Intermediate Maintenance Activity (IMA) Planning and Performance Accounting

2) Estimated Material Funding Requirements (7)

Requirements for material funding for IMAs are prepared and submitted both annually and quarterly by the TYCOMS. Annual requirements are generally for the entire TYCOM; quarterly requirements are by individual IMA. These requirements and LANTFLT quarterly allocations are prepared manually after the quarterly IMA schedule has been prepared. It was anticipated that later phases of the system implementation would interface directly with the IMA material program files to be incorporated into the CINCLANTFLT Financial Management Information System.

3) Quarterly IMA Schedule (8)

This file contains the schedule of IMA availabilities resulting from the IMA scheduling conference. It contains, by IMA, the planned arrival and departure dates of each ship serviced by the IMA. The schedule is sufficiently flexible to permit changes resulting from operational requirements, but it does permit the planning to ensure the best IMA utilization. Further, it ensures that ships with peculiar problems are supported by IMAs with special capabilities.

4) Personnel Requirements (9)

The data in this file result from reports from the TYCOM, IMAs, and the ships, which indicate that special skills and qualifications of personnel are either in short supply or are excess to requirements. This information is used to ensure that appropriate action is taken to acquire or transfer personnel to facilities possessing the requirements.

5) IMA Program Accomplishment

The data for IMA program accomplishment are currently reported under the 3-M system and is processed at DPSCLANT by the

Intermediate Maintenance Management System (IMMS). These data are reported monthly by each IMA.

6) Material Funds Expended (10)

This IMMS-derived data reports, by IMA work center and by ship tended, the dollar value of the material expended during the reporting period. Data elements include tender identification, work center, ship tended, and dollar value of material utilized.

7) Man Hours Expended (11)

The same reports which provide data pertaining to Material Fund expenditures also provide the report of man-hour utilization. 3-M data submitted by the IMA is processed at DPSCLANT in the IMMS system to derive, by work center and ship tended, the number of man hours expended. In addition, the file will contain the number of personnel available at each work center. Thus, the data in this file can provide insight to the degree of personnel utilization, by work center, during the period.

8) Schedule Adherence (12)

Each IMA currently reports the arrival and departure dates of every ship tended. Additionally, it reports the work accomplished and the status of the Current Ship Maintenance Plan (CSMP) for each ship tended. These reports are processed monthly at the DPSC using IMMS. These data are currently available for application to NAVLIS. Comparison of the data contained in this file with the data contained in the Quarterly Schedule file will show the degree of adherence to the schedule.

9) Intermediate Maintenance Program Effectiveness

The six data files of the IMA Planning and Performance Accounting module contain the basic data to permit analysis of the IMA program. As experience is gained in the use of the system,

more incisive results will be attained and more sharply defined data will be added to the files. The initial analyses will doubtless identify areas in which increased insight is desirable.

Analytic results expected to be attained initially from the data are:

° Adherence to, and departures from, the scheduled availabilities.

° Amount of unscheduled IMA availability and historical trends.

° Effectiveness of man power and material utilization of each IMA, and the relative effectiveness of each IMA when compared with all others.

° Status of the CSMP and comparison with previous data and indicated trends.

° Identification of problem areas such as manpower or material shortages.

° A basis for work load leveling among the IMA's.

(d) Investigation of the Organizational Maintenance Planning and Performance Accounting Module - Figure 5

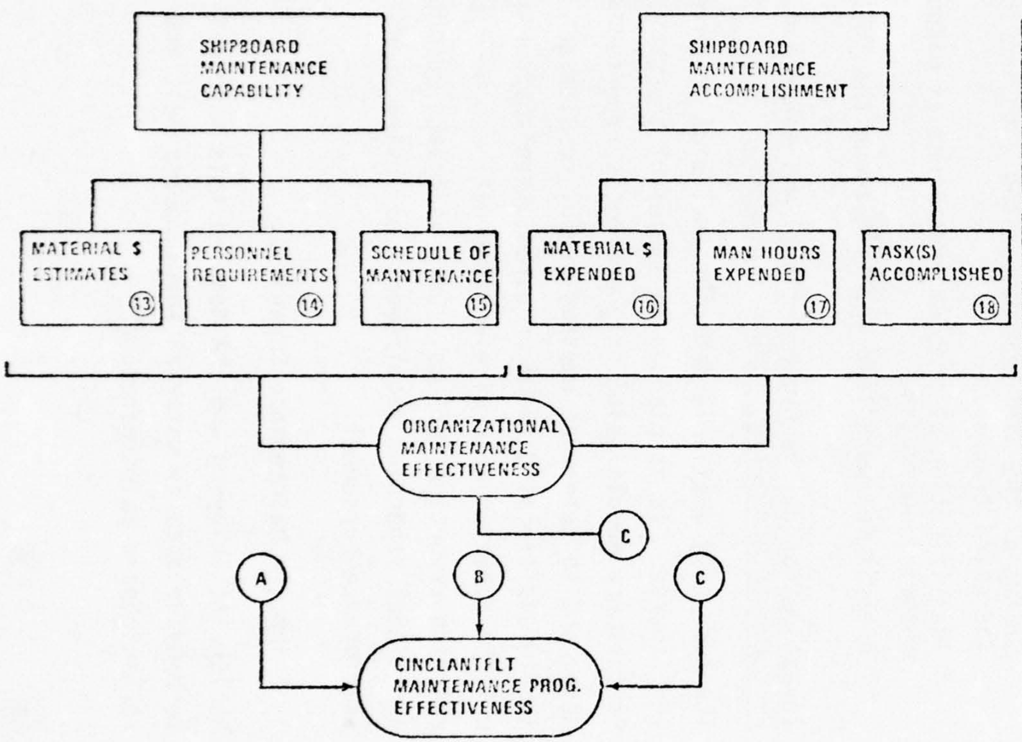
The currently available data for inclusion in this module are very sparse. It is planned to implement the OM Planning and Performance Module using only presently reported and available data, and to determine whether useful results can be obtained. Should additional data be required, then appropriate measures will be taken to determine whether data can be secured from existing reporting systems. Only as a last resort will additional reporting requirements be levied upon shipboard maintenance personnel.

1) Shipboard Maintenance Capability

As for the shipyard and IMA Modules this module is configured to provide data on material funds, personnel, and schedules for shipboard maintenance.

NAVLIS
CINCLANTFLT

ORGANIZATIONAL MAINTENANCE
PLANNING AND PERFORMANCE ACCOUNTING



NAVLIS INTERFACES

FUNCTIONAL FILE	PRESENT (PHASE I)	FULL NAVLIS (PHASE II)
13 MATL $ ESTIMATES	SHIP/TYCOM MANUAL	FMIS
14 PERSONNEL REQUIREMENTS	?	?
15 SCHED OF MAINTENANCE	?	?
16 MATL EXPENDED	3 M SUPPLY SYS MAN.	SUPPLY SYS
17 MAN HOURS EXPENDED	MANUAL	3 M + (?)
18 TASKS ACCOMPLISHED	?	3 M + (?)

Figure 5 - Organizational Maintenance and Performance Accounting

2) Material Fund Estimates (13)

Estimates of funds required for material to be used in the accomplishment of organizational maintenance are submitted by each ship to the TYCOM. The TYCOM maintenance and supply staffs make the necessary adjustments and submit them to the CINCLANTFLT Maintenance and Material Readiness Division, which in turn consolidates TYCOM estimated requirements for inclusion in the budget program. This file will contain the values of the funds for material programmed for each ship for the FY.

3) Personnel Requirements (14) and Schedule of Maintenance (15)

At this time sources for these data have not been identified. Exception data reported through the CASREP and 3-M Systems are potential sources. Summary data from IMMS can provide some insight to the CSMP.

4) Shipboard Maintenance Accomplishment

These files contain historical data on organizational maintenance accomplishment. Comparison of these data with the data describing the organizational maintenance capability provides insight into the effectiveness of shipboard maintenance.

5) Material Funds Expended (16)

Data in this file are derived from supply system reports which periodically report by ship and cognizant activity the dollar value of material furnished to each ship.

6) Man Hours Expended (17) and Work Accomplished (18)

Data for these files are not currently directly available. The 3-M System does provide for some data, such as exceptions reported for particular equipment systems, or repairs beyond the capability of the organizational maintenance resources.

It is believed that, with monor modifications, insight into shipboard maintenance accomplishment can be acquired from the IMA. For example, changes in the CSMP back log could provide information on personnel qualifications and numbers.

7) Organizational Maintenance Effectiveness Analysis

Initial measures of organizational maintenance effectiveness will be related to the expenditures for material and changes en the CSMP. Later analysis will be predicated upon man power availability and utilization, and on scheduled workload adherence.

8) CINCLANTFLT Maintenance Program Effectiveness

Output of the three basic modules, (1) Shipyard Maintenance Module, (2) IMA Module, and (3) the Organizational Module, will be configured to be directly comparable. Typical analyses will include:

- Relative costs of each level of maintenance
- Relative productivity of each maintenance echelon
- Comparative effectiveness of each maintenance facility
- Comparative maintenance system effectiveness for
  - Like ship types
  - Various ages of like ship types
  - Alternative maintenance strategies
  - Alternative maintenance intervals
  - Alternative funding strategies

4.2 DPSCLANT EFFORT

The Data Processing Service Center, Atlantic Fleet was preparing to implement the NAVLIS Executive System on its Spectra 70/45 in preparation for two major DPSCLANT efforts:

1) the incorporation of the Center as an operating node (site) on the NAVLIS Pilot Network
2) the establishment of an initial on-line capability at the Center to support the local TYCOMS and other interested (and funded) users.

The NAVLIS termination will undoubtedly have an adverse effect on the ability of DPSCLANT to meet at the second objective which would have contributed directly to the operational effectiveness of the Fleet.

5. PILOT MODEL STATUS AT PROJECT TERMINATION

At the termination of the NAVLIS project every major module in the NAVLIS system was undergoing significant enhancement. There were actually two separate NAVLIS versions. An "executable" system was installed and was being tested at the Naval Material Command Support Activity (NMCSA) in Arlington, Virginia, and the Data Processing Service Center for the Pacific (DPSCPAC) in San Diego, California. The second version was a "development" version with module debugging and modification occurring on the IBM 360/65 and 370/165 at NMCSA and on the UNIVAC Spectra 70/45 at the Navy Command Systems Support Activity (NCSSA) in Washington, D.C. The development version evolved partially in response to practical problems encountered while testing the executable version at DPSCPAC and NMCSA. Enhancements to the development version were in preparation for expansion of the pilot model network and included significant new capabilities, such as secondary hit resolution, multiple executives, increased hit resolution efficiency, and access to a non-DMS supported data base. Most of the coding required for these new capabilities had been incorporated into the modules at project termination, but module testing had not been completed and integrated testing had not begun. The status of both the executable and development versions is discussed in more detail below.

5.1 "EXECUTABLE" NAVLIS PILOT MODEL

This version of the NAVLIS Pilot Model system had been implemented and was being used for system testing at DPSCPAC in San Diego and NMCSA in Arlington, Virginia. The executive software was resident on DPSCPAC's UNIVAC Spectra 70/45, and the remote or satellite software was installed on NMCSA's 360/65 and 370/165. The executive software provided DPSCPAC with considerably more capability than a satellite site. An executive site had the following capabilities:

a. Processing user's log-on
b. Accepting and processing queries
c. Accepting and processing report definitions
d. Accepting user specified file parameters for limiting data base searches
e. Performing "file selection", i.e., determining which data bases in the network could respond to the query
f. Transmitting and receiving data from other network sites
g. Performing synonym resolutions
h. Performing hit resolutions
i. Accessing data bases and retrieving data
j. Accumulating responses from all sites
k. Sorting response data as required
l. Generating output reports

A satellite site had the software capable of performing only functions f, g, h and i above.

For this initial test version of NAVLIS, two data bases were accessible to users, the Aircraft Power Plant Management System (APPMS) at DPSCPAC and the Aircraft Engine Accounting (AEA) file at NMCSA. The APPMS data base was supported by the SHARP data management system at DPSCPAC which made the NAVLIS interface to that data base relatively simple as NAVLIS and SHARP have much retrieval software in common. The AEA data base at NMCSA was supported by IBM's IMS data management system. This system required a special NAVLIS interface to that data base, which was quite complicated.

A typical testing session involved the input of test queries from DTNSRDC to retrieve data from either or both of the systems data bases. A portable terminal and dial-up communications were used to establish contact with the executive site at DPSCPAC. Dial-up telephone lines were also used to establish communication between the two computers. Pre-test preparations required telephoning the operators at both DPSCPAC and NMCSA to bring the system up. DPSCPAC then made the dial-up connection between the Spectra 70/45 and the 360/65 (or 370/165). The user

then dialed in to the DPSCPAC Spectra 70/45 to make the terminal connection to the executive software package. With communication established, the NAVLIS System was ready to accept queries or report definitions. The queries were designed to test as many of the system features as possible. The inevitable "bugs" discovered were rectified in the "development" modules, unless they significantly impacted the testing process, in which case they were corrected in both the development and the executable versions.

Query input was accepted and edited by the Query Translation Module (QTM) at the "executive" site. Once the query had been received and validated, the File Selection Module (FSM) interfaced with the user to determine whether the user wished to supply file descriptors or parameters to limit the data retrieval to certain types of data bases. FSM then identified those data bases in the NAVLIS System that were candidates to respond to the user's query and satisfy the file parameters the user provided, if any. FSM generated copies of the query to go to each satellite or remote site containing a candidate data base. The Executive Control Module (ECM) and the NAVLIS Telecommunications Subsystem (NTS) effected the transmission of the queries to the appropriate sites. The Pilot Model, as it existed at project termination, had only one remote site, at NMCSA. In practice, the queries normally required accessing the AEA file at NMCSA and the APPMS file at DPSCPAC.

After a copy of the query was transmitted to NMCSA for AEA file access, with a copy retained at DPSCPAC for APPMS file access, the following actions occurred simultaneously at both sites. The Synonym Resolution Module examined the Synonym Resolution File and the query to determine whether any search acronym or value translations were required to make the expression of the query compatible with the contents of the data base. Following any such translation, if required, the Primary Hit Resolution Module (PHRM) examined the query constraints and the inverted files associated with the data base to be accessed to identify those records that satisfied the query constraints and contained the data elements requested. The keys for these records were written to a temporary file and passed to the ADS' Interface Module, which accessed the appropriate data base, retrieved the

requested data values, and formatted those values into "response" records.

The response records generated at the executive site (DPSCPAC) were written directly to a temporary Response File. The response records generated at a remote site (NMCSA) were passed to the NAVLIS Telecommunications Subsystem (NTS) for queuing until all responses had been generated. The queue was then transmitted to the executive site where it was appended to the Response File. This file, which now contained all the data retrieved to answer the user's query, was manipulated by the Sort Module (SM), if the user has specified a sort sequence for his output. The Report Generator Module then accepted the sorted response records and formated the output according to either a default format or a pre-defined report format specified with the query.

5.2 "DEVELOPMENT" NAVLIS PILOT SYSTEM

The NAVLIS Pilot Model under development at project termination was essentially the executable system, described briefly above, with significant additional capabilities. The Pilot Model was in the process of being expanded from a two-node network (DPSCPAC and NMCSA) to a three-node network with the addition of the Data Processing Service Center for the Atlantic (DPSCLANT) in Norfolk, Virginia. DPSCLANT was to be a second "executive" site, so that a user could make a terminal connection at DPSCLANT or DPSCPAC, whichever was most convenient. Each of these two sites was to have the full range of executive capabilities (a through l above) and could retrieve data locally or from either or both of the other two facilities (NMCSA and DPSCPAC, or NMCSA and DPSCLANT). When either DPSCPAC or DPSCLANT was to perform the executive function for a particular query, the system would recognize that condition and insure that the other executive site performed only the satellite or remote site range of functions for that query. For example, if the user made a terminal connection to DPSCLANT and input a query that accessed data bases at both NMCSA and DPSCPAC, the DPSCPAC software would recognize that only the remote site functions (synonym resolution, hit resolution, and data retrieval) were to be performed for this query, and that the

executive type functions (query translation, report generation, etc.) were being performed by another node in the system. The development of the multi-executive capability in the network would have provided much greater flexibility in determining the location of sites that could most efficiently control the the system retrieval operation.

As mentioned previously, both the data bases accessible by the "executable" NAVLIS system were supported by data management systems (DMS's). With the addition of DPSCLANT to the network, a non-DMS supported data base would have become accessible through the system. The project's intention was to mount the DPSCLANT counterpart to the DPSCPAC APPMS data base as an indexed sequential file. As an ISAM data base, the application programs that were currently processing the file sequentially could continue to do so. Additionally, the NAVLIS system could have accessed the file randomly. This completed, NAVLIS would have been able to demonstrate the ability to access a data base supported by a compatable DMS (SHARP), to interface with and access a data base supported by a noncompatible DMS (IMS), and to interface with and access a non-DMS supported data base, simultaneously, at remote locations, in response to the same query.

The secondary his resolution capability was to be incorporated into the system concurrently with the addition of DPSCLANT to the network. This capability would have permitted the user to query on any data element in the system, regardless of whether that element had been selected for inversion (without the user having to know which elements were inverted and which were not). This capability is found in few data management systems. Further discussion of the secondary hit resolution capability can be found in Appendix A. At project termination the "executable" version of the system included only the primary hit resolution capability, which required all the query constraints to be on inverted data elements. Retrieval of information, however, could be performed on both inverted and non-inverted data elements.

Several of the development modules, the ADS Interface Module and the Primary Hit Resolution Module in particular, had undergone significant modification allowing them to process considerable faster and more efficiently than their "executable" system counterparts. Several other system efficiencies had been incorporated into the development module, along with corrections to coding, design, and data errors discovered while testing the "executable" system. All the capabilities discussed above were due for implementation in the NAVLIS Pilot Model within two to three months of the date on which the project termination.

6. NAVLIS MANAGEMENT CONSIDERATIONS

The technical aspects of the NAVLIS project (developing a logical development sequence from pilot model to proposed implementation) represented but a portion of the task of successfully applying and implementing such a system.

The management of a NAVLIS-type application would be critical to its ultimate effectiveness. The experience gained from developing the NAVLIS Pilot network, even though that work was not completed, suggested the following managerial considerations:

a) the management of a NAVLIS would require a staff of logisticians and systems analysts at the Headquarters (OP-91, OP-04) level to make policy and set priorities.

b) at least two analysts, with joint responsibility for assuring that NAVLIS files and software were kept operational, would be required at each network site.

c) stringent software configuration control procedures, such as the ones employed during the NAVLIS pilot development, would have to be initiated.

d) a User Assistance Group would be required to process user requests for system changes and to educate new users.

e) standardization of nomenclature among data systems for use in describing like data elements would be useful, if not necessarily a requirement.

f) primary and secondary sources of similar information would have to be identified and incorporated into an intelligent data directory.

g) user access procedures and restrictions would have to be developed at the file, record, and data element levels to safeguard confidential information (data).

h) built-in safeguards to protect against the construction of classified information from segments of data obtained from various data sources would have to be inserted in the control software.

i) control of updating procedures and cycles would have to be imposed to assure the consistency of data throughout the network.

7. SUMMARY OF PROJECT EXPENDITURE BY DTNSRDC

The following is an outline of the expenditures and purposes for those expenditures on the NAVLIS project since its transfer to DTNSRDC in FY 73.

7.1 FISCAL YEAR 1973 EXPENDITURES

Total Funding - $60K

Expenditures

Amount		Purpose
$31.8K	-	Labor/Overhead (DTNSRDC-Inhouse). Review and evaluate current status of NAVLIS and make recommendations as to its disposition and direction.
.5K	-	Misc: (Computer, Report Publishing, etc.)
10.5K	-	Contract to ManTech of N. J. to organize and classify NAVLIS Technical Library.
* 17.2K	-	Return to NELC
Total $60.0K		

7.2 FISCAL YEAR 1974 EXPENDITURES

Total Funding - $312K

Expenditures

Amount		Purpose
$160.1K	-	DTNSRDC - Inhouse Develop Preliminary Design Spec for NAVLIS Executive. Program Management, Pilot Configuration Determination.
14.6K	-	Computer Charges/Travel
8.5K	-	NELC to digitize CDDB for Transfer to DTNSRDC
49.6K	-	ManTech of N. J.

69.7K	-	Rockwell International: for Scenario Development Detail Functional Spec for Selected Executive and T/P modules.
25.0K	-	Data Processing Services Center - Two months of hardware upgrade rental in preparation for NAVLIS development and participation.
Total $327.4K		

7.3 FISCAL YEAR 1975 EXPENDITURES

Total Funding - $1.011M

Expenditures

Amount		Purpose
$354.2K	-	In-house DTNSRDC, Pilot Executive Software Development, Exec. Control Module, Query Translation Module, Sort Module, Primary Hit Resolution Module, Report Definition Module, Report Generation Module, etc.
10.8K	-	Prior Commitment (Rockwell SPCC)
59.5K	-	Computer Time and Travel
3.6K	-	Returned to Sponsor
48.0K	-	DPSCLANT, Hardware upgrade to prepare for Pilot participation
5.0K	-	Naval Weapons Engineering Support Act.; Scenario Dev.
2.3K	-	NMCSA, Rental of Sync. Line Adaptor to allow use of TSO in 360/370 Software development.
108.2K	-	NAVCOSSACT, NAVLIS TELCOM Subsystem Development, CPU-to-CPU Software
215.0K	-	DPSCPAC $150K Hardware Rental 65K Software Support
6.0K	-	SPCC Misc request S-M Data File Generation

	55.2K	- Rockwell Int'l Data Analysis Module Spec Self Management Module Spec CINCPAC Requirements Analysis Spec
	30.0K	- Withdrawn by Sponsor
	15.4K	- Deficit from FY 74
	103.3K	- ManTech of N. J. Cost/Benefit Methodology
Total	1016.5K	

7.4 FISCAL YEAR 1976 EXPENDITURES

Total Funding - $1.040M

Expenditures

Amount		Purpose
$ 95K	-	NAVCOSSACT, T/P Support Pilot Network between DPSCPAC and NMCSA T/P Alternatives Study
13.0K	-	NAVCOSSACT (NAVTELCOM) TELCOM COMPUTER COSTS
75.0K	-	DPSCPAC, Hardware Rental, Analyst Support (minimal)
75.0K	-	DPSCLANT, Hardware Upgrade Rental
.2K	-	CINCLANTFLT, TELCOM Cost for NAVLIS Appl.
5.0K	-	FMSO, 3M Data Generation
11.6K	-	Rockwell International (Prior year adjustment)
12.0K	-	General Electric Software Dev. Support
55.0K	-	ManTech, CINCLANTFLT Requirements Analysis

	241.6K	-	In-house DTNSRDC Software Development
	42K	-	Final Program Documentation Recommendations/Spec.
	414.6K	-	Returned to Sponsor upon project termination
Total	1040.0K		

8. TECHNICAL CONSIDERATIONS AND RECOMMENDATIONS

This section of the report will identify, through recommendations, areas in which the NAVLIS concept could have proven beneficial to the Navy, and will indicate, through considerations, several benefits and problems associated with system design characteristics unique to a NAVLIS-type effort.

8.1 CONSIDERATIONS

The considerations discussed here touch on the salient aspects of the NAVLIS technical effort in the Executive Subsystem. The techincal considerations are as follows:

8.1.1 GENERALIZED "ADS INTERFACE" CONSIDERATIONS

Of particular concern, early in the design phase of the NAVLIS system, were the difficulties posed by the requirement to read multiple data bases of dissimilar format and organization. The data files identified as likely candidates for NAVLIS access included sequentially organized files designed for and operating in a batch processing mode, as well as data files maintained by large and powerful data management systems with their sophisticated data base structures. The effort and overhead associated with generating unique software programs to access and extract data from each of the data files to be included in the system appeared to be excessive. However, the feasibility of developing a single program capable of accessing all the systems data bases was questionable. Since the initial NAVLIS efforts would be dealing with a limited number of data bases, the option to develop a unique program (the ADS Interface Module) to interface the system with each data file was selected. It was anticipated that the experience gained by interfacing each data base with its own module would provide some bases for determining whether a generalized ADS Interface Module was feasible and/or practical.

The two most important functions of the ADS Interface Module are to read records from its associated data base and to extract from those records the data values requested by the query. During the development of the module to accomplish those functions, it became apparent that, by using table structures and subroutines, a generalized ADS Interface Module was feasible and would be more practical than the file structure dependent versions which had been used for system development. The ADS modules currently use a table structure to define the format of the data record. For each data field that can be retrieved from the record, the table contains an entry that indicates the position of the first character in the field and the number of characters in the field. For fixed length/fixed field records, this information remains constant throughout module execution. For variable length records and fields, the positional information is entered in the table after each record is read and before value retrieval has begun. The data required to determine the location of each field in the record is normally contained in a fixed segment of the record itself. Data files containing multiple record types, each with a unique format, present no problem other than the need for multiple tables to define each record type.

To generalize the ADS Interface Module, a file could be constructed at each site in the network containing the record format(s) (in terms of field positions and lengths) for each data file to be accessed at that site. Prior to accessing a specific data base, the information needed to complete the table that defines the record structure of the data base would be read from the file of record formats at that location. This represents a significant step toward ADS Interface Module generalization by removing the necessity for hard-coded record descriptions in the module.

The statements required by the ADS Interface Module to actually perform the read function may differ significantly as a result of the access method employed, the record key specifications, and the data management system, if any, supporting the file to be read. By assigning the reading process to a unique subroutine, the ADS Interface Module achieves another level of generalization. All file peculiar processes related to the read function could be performed by these subroutines,

e.g., opening and closing the file, error conditions, and end-of-file conditions. Other file peculiar functions directly associated with the reading of the record, such as those that would be available through a data management system, would be concentrated in the subroutine associated with the data file to be accessed. Functions performed by a subroutine that accessed a file supported by IMS, for example, would include modifying the "GET" operator in the IMS "CALL" statement to effect the desired type of segment retrieval.

An additional area of processing that is file peculiar involves data value conversions. Although NAVLIS requires no data alteration on a file to be accessed, there are conditions in which a retrieved value must be converted to a common system form of representation. A NAVLIS "standard" is required to achieve a degree of output consistency, as well as to permit arithmetic operations on retrieved data. For example, quantities for the same item may be recorded in different units of measure on different data files. The discrepancy of recording quantities in tens of units and hundreds of units must be resolved for both consistency of display and data manipulation. In many cases, the conversions required are unique to a specific data file at a site. Such file peculiar conversions could be performed by the subroutine that executes the access functions for that data file. When several files at one site require the same type of conversion for some of their data values, e.g., Julian date conversion, the coding required to perform that function could be included in the ADS Interface Module.

In short, the degree of commonality that emerged among the ADS Interface Modules developed for the NAVLIS Pilot Model suggests that a unique interface module for each data file is not a requirement. A considerably more efficient approach would be to utilize a single ADS Interface Module to perform common functions, e.g., determining data elements to be retrieved, generating response records, interfacing with the control module (ECM or RCM), performing common conversions, etc., and to use subroutines for file accessing and file peculiar data conversions. A critical feature for this approach would be a system support file accessable to the ADS Interface Module containing the record

format data, expressed as data field starting positions and lengths. The generalized module would then execute by reading the support file and initializing the record format table with the entries describing the records on the data file to be accessed. The ADS Interface Module would execute those functions common to all data retrieval regardless of the data file being accessed, and call subroutines to execute file peculiar accesses and conversions.

8.1.2 SYNONYM RESOLUTION CONSIDERATIONS

In the development of both the San Diego demonstration and the Pilot model, some of the difficulties of dealing with nonstandardized data elements were readily apparent. Much of the disparity could be rectified by performing conversions as data were extracted from the record. For example, Gregorian or Julian date conversions are easily performed by a small number of programming statements. The conversion of units of measure to some internal system standard is also performed easily.

Whenever a definable and consistent algorithm can be applied to translate the value being extracted from the data file to an internal standard representation, the conversion required can be performed with little difficulty (assuming a reasonable algorithm). However, when an algorithm cannot be defined to perform a required conversion, the problem becomes more complicated.

This situation presents itself when different data files use different conventions for identification of the same item. A ship, for example, may be identified by name in one data file, by Unit Identification Code (UIC) in another file, and by Permanent Unit Code (PUC) in a third file. The problem is complicated further when the file using the name of the ship for identification has no established procedures for recording that value, or when the procedures are established but not followed. As a result, the "ship name" field may contain several different values referring to the same vessel, i.e., U.S.S. Midway, USS Midway, Midway. Assuming a query is to be generated

requesting data related to a specific ship, the terminal operator could identify that ship by any of the above types of value or by an entirely different value type, e.g., the ship's hull number.

In a retrieval system capable of accessing multiple data files, all data relevant to the query should be retrieved regardless of different identification conventions. Both the Pilot model and the San Diego demonstration addressed this problem through the use of the Synonym Resolution Module (SRM). A Synonym Resolution File was created for each data file to be accessed and was to contain the item identifications as they were recorded on the associated data file. To continue the example above, if "Ship Name" was the field of concern, the Synonym Resolution File would contain each ship name that appeared on the data file. Associated with each ship name on the Synonym Resolution File would be equivalent identification values, and their type, that were to be accommodated. The Synonym Resolution File would then indicate the equivalency of "Ship Name", "UIC", "PUC" and "Hull Number" as data types for ship identification, and would also indicate what specific value in each type would identify a particular ship. This relationship would be indicated:

USS ABCD (SHIPNAME) = 12345 (UIC) = 67890 (PUC) = ABC-12 (HULL NO.)

Using the Synonym Resolution File, the SRM translates the query input into the equivalent data type and values necessary to access the associated data file. If a data file identifying ship by UIC code is to be accessed, a query input of

IF SHIP-NAME IS USS MIDWAY

would be translated by the SRM to the internal system equivalent of

IF UIC IS 03341

The same translation is performed for data requested by the query.

PRINT SHIP-NAME becomes PRINT UIC

In the Pilot Model and the San Diego demonstration, the SRM capability was used to translate ship names, organization names, and aircraft squadron designations to their equivalent UIC and PUC codes and vice versa.

Also accommodated were various permutations of names and designators to allow the user more flexibility in query formulation. For example, equivalency was established between "COMNAVAIRPAC" and "CNAP", and between "CVA-41" and "CVA 41". Care was taken to select commonly used abbreviations and permutations to prevent the size of the Synonym Resolution File from becoming too large.

The SRM was partially successful in enabling access to multiple data files despite encountering different values for identification of the same item. At project termination the SRM performed only the synonym resolution required to access the data file; the module did not translate the retrieved value into what the user had originally requested. If a user had requested the retrieval of a ship's name from a particular data file and ships were identified in that file by UIC code, that code would be retrieved and displayed to the user. If synonym resolution is to be performed, it would be beneficial to translate the data retrieved into the form originally requested by the user, or to advise the user that synonym resolution had occurred and identify the values being retrieved, e.g., names, codes, etc.

Although informing the user of the translation that has occurred is beneficial, it does not overcome certain problems inherent in the substitution of one value for another, especially as the sophistication of the query increases. One such difficulty becomes apparent when a query specifies the retrieval of ship name, as well as other values, from records that satisfy the query constraints and also requests that the output be sorted on ship name. When codes are retrieved from one file and names from another, the execution of the sort is not likely to produce the result the user had intended.

When the query language used allows the user to specify "prefix", "suffix" or "partial" value searches, as did the NAVLIS Pilot Module, the substitution of a different value as a result of synonym resolution can produce meaningless results. Similarily, the use of such conditions as "greater than" and "less than" can present problems. In this case, the substitution of an alphabetic name value for a numeric code value

could conceivably produce retrieval of data elements from every record on the data file.

The problem of synonym resolution is a difficult one when trying to allow simultaneous access to multiple data files. Whenever possible the system should resolve differences in identification conventions between files without requiring the user to input several values. However, the system should also be able to determine when value substitution may produce meaningless or erroneous results.

Further development of the synonym resolution capability should attempt to solve these problems. It is probable that an examination of the query structure by the SRM could identify most cases in which value substitution might not produce the intended results. The module might then begin a dialogue with the user identifying the problem and potential hazards and alternatives, if any.

8.1.3 APPLICATIONS VS AD HOC QUERIES CONSIDERATIONS

Much of the emphasis during NAVLIS system design and development was placed upon the ability of the system to respond to ad hoc queries generated by logistics managers. This emphasis was a natural result of the "demonstration" oriented growth of the system. The capability to respond to such queries is an important one, but an equally important capability is in using the simultaneous, remote multi-file access facility to perform regular and repeated applications processing.
At project termination an effort was underway to develop an application package for ship maintenance and repair for CINCLANTFLT. Earlier in the NAVLIS project, a demonstration of the usefulness of this type of system in solving existing reporting and operational problems was requested by COMNAVAIRPAC. In conjunction with logistics managers and Data Processing Service Center - Pacific (DPSCPAC) personnel, the Aviation Consolidated Allowance Test/Fleet Program Support Material (AVCAL/FPSM) problem was identified as relevant to a NAVLIS type system. This experiment is documented in detail in "Experiment Documentation for San Diego Demonstrations" produced by Rockwell International on

28 June 1973. Briefly, however, the problem involved determining additional spare part support requirements at NSD Subic Bay when the weapon system of an aircraft carrier in the western Pacific changed by aircraft quantity, configuration, or type. At that time, and undoubtedly at this writing (two and a half years later) the process for solving that problem involved mailing three magnetic tapes from ASO, Mechanicsburg, to DPSCPAC where a series of tape conversions and batch programs was run to produce a card deck. The card deck was then mailed to NSD Subic for processing and production of another card deck and a report which were mailed to AIRPAC and SERVPAC, respectively. When additional procurement was required and approved by SERVPAC, AIRPAC and NSD Subic were notified by mail. This process required anywhere from days to weeks with all the attendent problems of tape and card handling and mailing. In an attempt to show how the operation could be performed in a NAVLIS environment, the facilities involved were simulated on a commercial time sharing network and subsets of the operational data files were mounted on that network. With the capabilities inherent in a NAVLIS-type system to initiate processing at remote facilities, retrieve data from remote facilities to be used as input to local processing, and transmit output to other remote locations, an AVCAL/FPSM problem could be solved in several minutes or hours depending upon the complexity of the problem.

When data from another location are needed to perform some reporting or operational function, the typical process, as with the AVCAL/FPSM example, is to copy and mail tapes from one site to another periodically. In some instances, copies of files or subsets of files are sent to several locations for processing. When this process involves a data base that is subject to much change, the mailing must be frequent or the data rapidly become obsolete. A considerably more efficient approach would be to retrieve only the data necessary for a specific process and only when that process is to be executed.

For example, assume multiple, service-wide reporting and operational requirements that utilize 3M data. Currently, facilities satisfy that

requirement by requesting copies of certain subsets of the 3M data base, or, more frequently, by creating and maintaining their own subset of 3M data. The first alternative creates an additional workload for 3M data base personnel and provides the user with data that daily become more obsolete. The second alternative usually involves maintaining a copy of data input to 3M, generating additional work for personnel at the user's facility and contributing to multiple copies of the same data. If users could retrieve, during or just prior to program execution, only the data relevant to the program from a single master data base, the problem of obsolete data, duplicate data bases, duplicate data base maintenance efforts, and additional workload at both facilities would be alleviated.

The NAVLIS system held the promise of accomplishing these objectives. Among the NAVLIS goals were the capability to pre-program the remote retrieval of specific data elements from both local and remote data bases, and the ability to generate temporary files from that retrieved data for input to local programs, to initiate specified programs automatically, and to transmit output to local and remote terminals. This combination of capabilities would enable request submission, data input, program processing, and output printing to be done at four different locations, if required.

8.1.4 USER PARTICIPATION IN RETRIEVAL SELECTION CONSIDERATIONS

The NAVLIS objective of providing simultaneous access to multiple data bases at several geographically remote facilities immediately presented problems concerning the control of the retrieval process. Requiring the user to be aware of which facilities and which data bases contained the values necessary to respond to his query was unacceptable. At the same time, without the capability to restrict the search under certain conditions, the volume of response data could become unmanageable. During NAVLIS Pilot Model development and testing, programmers and analysts who were well acquainted with system capacities and data base contents inadvertently generated queries whose responses overflowed

the system's files. The problem would become significantly more acute with users unfamiliar with the contents of the data bases in the network. Some provisions for limiting the search area at query time and for detecting and preventing impending overflow conditions were mandatory.

The NAVLIS Pilot Model currently has the capability of determining, without user intervention, those data files that could satisfy the user's query and of performing data retrieval from those files. Early in the NAVLIS design phases it was recognized that some system users would be sophisticated enough to identify a subset of the system's data bases that was relevent to their query. A capability was provided, therefore, to allow the user to enter certain descriptions or parameters prior to the selection of candidate data bases by the system software. When such parameters are input by the user, e.g. "AIRCRAFT", "FIRST FLEET", "MAINTENANCE", etc., only the data bases satisfying those parameters are considered for retrieval.

Parameters describing the types of data contained in a base were manually assigned when the data base was incorporated into the network. The name of the data base was also one of the parameters assigned enabling a user to designate a specific file(s) for retrieval. (A beneficial enhancement, not currently a feature of the Pilot Model, would be to identify to the user those data bases that satisfied his parameters and allow the user to accept or modify the list of candidate data bases or input a new set of parameters if desired). This capability would provide some user control of the retrieval process, but assumes some knowledge of file content by the user, and still provides no guarantee against excessive retrieval.

Although the data bases that might satisfy the user's query are identified early in the processing cycle (by the File Selection Module), not until after hit resolution has been performed is there an indication of the amount of data that will be retrieved from a data base to respond to the query.

The current Pilot Module is subject to file overflow, by exceeding allocated internal table capacities, and program termination in several areas:

1) Hit File overflow - as the Primary Hit Resolution Module (PHRM) determines which records in a data base satisfy the user's query, the access key of each "hit" record is written to the Hit File.

2) Teleprocessing Message Queue overflow - as the ADS Interface Module at a remote site generates response records (one response record for each data element requested and contained in each hit record), they are passed to the teleprocessing module for queuing until all responses for that site have been generated. After all retrieval has been completed at a remote site, the response records produced are transmitted to the query site.

3) Response File overflow - the response records generated by ADS Interface Modules at the user's site are written directly to the Response File (no teleprocessing of this data is necessary). In addition, all the response records generated by and transmitted from remote sites are written to the Response File.

For each of the conditions above, it is possible to determine beforehand that an overflow will occur and to initiate procedures to prevent the overflow. The Primary Hit Resolution Module need only be provided with the maximum capacity of the Hit File and be asked to monitor the number of records being written to that file. On detecting an impending overflow condition, the module could suspend normal processing while determining what preventive action to take. The user should be advised, at this point, that at least one data base to be accessed will produce a very large number of responses. The user could then elect to exercise such options as complete query cancellation, partial retrieval from the data base affected, retrieval from data bases not indicating overflows only, etc. Procedures should also be developed to use tape files to accept the Hit File overflow if the user elects to retrieve all response data regardless of volume. In this event, procedures to prevent overflow

conditions on the teleprocessing queue and the Response File should automatically be initiated.

Under the current system design, all response records generated from the data base(s) accessed at one remote site are queued by the teleprocessing modules. When retrieval from all data bases to be accessed at a site is complete, that queue is transmitted to the user's site. Since each data base record access key can produce multiple response records (one response record is created for each data element requested by the query for each "hit" record that contains those data elements), and since the responses from multiple data bases at a site may be queued before transmission begins, the teleprocessing queue could exceed capacity even though the Hit File(s) involved were well within capacity. This condition did not occur during Pilot Model development and testing, largely because queries producing an excessive amount of retrieval would first cause a Hit File overflow condition. However, the Hit File allocation used for the Pilot Model was considerably less than would be required for a larger system. Additionally, if procedures are incorporated to accept Hit File overflow and continue processing, then procedures for accommodating teleprocessing queue overflows become mandatory.

As with Hit File overflow, the use of a tape file back-up to accept teleprocessing queue overflow is a possibility. A more efficient approach, however, would be to perform multiple transmissions. As the teleprocessing queue reaches capacity, response record generation could be suspended while data on the queue are transmitted to the user's site. After the queue is emptied, the creation of the response records can continue. This approach provides unlimited response record handling capacity for the teleprocessing queue. The overflow problem then moves further downstream in the processing cycle to the Response File.

The Response File in the Pilot Model resides at the user's site and is used to collect all responses generated for the query at all sites. As such, it is a prime candidate for overflow problems, especially as procedures are developed to enable the Hit File and teleprocessing queue

to accommodate more data. As with the Hit File, a tape back-up for the Response File has the advantages of ease of implementation and large capacity.

One significant enhancement that would do much to prevent unwanted excessive data retrieval would be to advise the user of the number of hits for his query prior to data retrieval. Currently, the Pilot Model advises the user of the number of hits and provides the option to terminate the query prior to report generation but after the response data have been retrieved. Ideally, the user should be notified of the volume of hits determined for the query immediately after hit resolution has been completed and before retrieval has begun. A Primary Hit Resolution Module (PHRM) resides at each site in the network. The module executes only once for each query and determines the number of hits from each candidate data base at that site. Processing for that query could be suspended at that point and the number of hits for each data base at that site transmitted to the control module at the user's site. PHRM at the user's site, if that site has data bases that are candidates for retrieval, would give the hits per data base directly to the control module. The control module would accumulate the hit information from each site and present it to the user. The query could be halted at this point giving the user the option to terminate, to continue with the hits identified, or to retrieve data from selected data bases only. The control module would then notify the appropriate sites to terminate or continue processing as specified by the user.

Whether or not these suggestions for controling data retrieval and file overflow survive a more extensive analysis of alternate methods, the problem of inadvertent, excessive data retrieval must be addressed in this type of system. As the network expands and more sites and data bases are added, the problem becomes more acute.

8.1.5 DATA FILE UPDATE CONSIDERATIONS

One of the constraints under which the NAVLIS Pilot Model was developed was that operational Navy logistics data files would be used without altering the structure of those files. Most of the data files

that were initially to be incorporated in the Pilot Model were sequentially organized data files designed for batch processing. The NAVLIS approach to accommodating data files with no existing direct access capability was to place those files on disk in sequential order, generate inverted files (indexes) to logistics data elements to be used for direct access, and modify the job control statement for applications programs that read those files to indicate that they were now on disk. In addition to using the Indexed Sequential Access Method (ISAM), or its equivalent, for retrieval, these procedures accomplished the objectives of minimum impact on both existing file structures and applications programs and provided a random access capability for acceptable response time to user's queries. This was essentially the approach taken at the Data Processing Service Center for the Pacific (DPSCPAC) at San Diego for the Aircraft Power Plant Management System (APPMS) data base. At the Naval Material Command Support Activity (NMCSA) the Aircraft Engine Accounting (AEA) file was supported by IBM's IMS data management system and had an existing random access capability. However, as the data base was originally structured for a purpose other than NAVLIS, access based on the logistic data elements of interest to NAVLIS was not provided by IMS. To provide that access, the AEA was inverted on the data values of interest to NAVLIS.

While creating inverted files provided random access not available previously, as in the case of the APPMS file, and via data values not accommodated previously, as with the AEA file, a significant update problem was presented. To be of any use, the inverted files had to stay in synchronization with the data file. This required that the inverted files be updated whenever a value of an element in the data base that had been inverted changed. The actual update of the inverted files was little problem, as the files and the update programs were peculiar to NAVLIS and had no impact on other programs. Identifying when values were changing, which values and in which record, was entirely different, especially for the IMS supported data base. In an uncomplicated update procedure, where all changes against the data files are represented by a tape or deck of transaction records (adds, changes, and deletes) that

are batch processed, the creation of a program to generate transactions against the inverted file from the same data base transactions on tape or deck is an efficient solution with no impact on the user's program. The maintenance procedures for an on-line data base supported by a data management system, however, are much more involved. In the case of the AEA file, update transactions occurred in both batch and on-line modes and the maintenance was performed by IMS. The most efficient and effective way to identify these transactions was to develop an interface between NAVLIS and IMS. Briefly, the interface intercepted the user program call to IMS and saved the data necessary to determine what action was occurring against the data base. The resulting file of activity against the data base was later processed by a NAVLIS program to update the inverted file for the AEA data base. This approach had the advantages of minor impact on IMS and required no change to any of the user's programs. A comprehensive discussion of the NAVLIS-IMS interface appears in Appendix A.

The point here is to identify the difficulty that was developing in the attempt to maintain the inverted files necessary for random access current with the latest version of the data base to be accessed, without significant modifications to the applications programs that used that data base. As the structure of the logistics data bases and the programs that maintain those data bases become more sophisticated, the interface becomes more complicated.

Early in the design phase consideration was given to much more sophisticated and efficient methods of providing random access to DMS supported data bases by using the indices (directories, inverted lists, etc.) used by the DMS itself to effect random access, provided the DMS indices included the logistics element relevant to a NAVLIS user. This approach would eliminate the inverted file storage space overhead as well as the problem of maintaining the inverted file (assuring the DMS indices were kept current with data base changes).

Another alternative that would have eliminated the use and maintenance of NAVLIS inverted files was to perform a translation for each DMS supported data base that would express the NAVLIS retrieval requirements in a format understandable by the native DMS. The DMS would perform the

retrieval functions using its own indices and the output would be reformatted to be compatible with responses from other data bases.

Both of these approaches would have served to eliminate the need for the NAVLIS inverted files. The disadvantages, however, appeared to far outweigh the benefits. The software development required to interface directly with the indices maintained by a data management system such as IMS is considerable. Once developed, the stability of that software is largely affected by the stability of the associated DMS'. As new versions, enhancements, and modifications to the DMS are incorporated, the probability of corresponding modifications to the interfacing software is high. The alternative of translating NAVLIS requirements into DMS statements limits the NAVLIS capabilities for that data base to those of the supporting DMS. In either case, the interface is difficult and places the retrieval function of the system in a position of dependency upon foreign software packages.

The relative ease with which transactions against the IMS data base could be captured and used to update the NAVLIS inverted files obviated the need for one of the more difficult interfaces above. Additionally, the incorporation of the secondary hit resolution capability, allowing retrieval on noninverted data elements, drastically reduced the inverted file storage overhead.

8.1.6 DATA ACCESS RESTRICTION CONSIDERATIONS

Handling classified data and data considered proprietary by its user organization was an early NAVLIS problem and had yet to receive adequate attention when the project was terminated. Since the Pilot Model was to use dial-up lines and non-secure facilities, unclassified files were selected for use and the problem was avoided. It was well recognized, however, that an operational NAVLIS network would be required to accommodate classified data or its usefulness to logistics managers and planners would be significantly reduced. Instructions for handling classified data are explicit concerning requirements for system components, e.g., data transmission, terminal access, etc. One of the difficulties peculiar to a NAVLIS-type system, due to the multi-file access capability, is the

inadvertent creation of classified information. The ability to collect otherwise remote pieces of information presents the possibility of generating classified material by aggregation. Only when the aggregation could be pre-defined could system safeguards be incorporated to recognize when the collection was being compiled. For example, when various types of data are susceptable to classification by aggregation, the NAVLIS system could recognize the possible generation of classified data. The File Selection Module uses the contents of the systems data files, identified by data type, to determine which data bases are candidates to respond to the user's query. It is at this point in the processing that the system could detect the possible generation of classified information by combining certain data types. However, this approach could generate many false alarms. The user could be informed of the possibility of generating classified information with his query, but the value of such a warning is questionable. It would do nothing to prevent intentional unauthorized retrieval. Another option might be to notify a system security organization which could determine the classification of the output generated prior to its receipt by the user and also determine the user's authorization for access to that data.

The question of handling and even identifying classified data in a NAVLIS system needs considerably more attention. There is, however, another level of security of concern to organizations which allow their data bases to become part of a data sharing network. The issue is not one of sharing classified data, but of releasing "unsanitized" data. In many cases, information generated from a data base is reported to higher command levels only after it has been "approved" at lower levels. Lower and middle level managers are concerned about the exposure of upper management to raw data. Often there are legitimate reasons for this concern. Frequently data values need some manipulation or should be considered in combination with other data elements to be meaningful and accurate. The total "quantity-on-hand" of an item could be very misleading without consideration of the status of the individual items. The retrieval of this type of accurate but incomplete data is a legitimate concern.

As with the classification problem, the system software can provide protection against incomplete retrieval only to the extent that it can be defined. When data elements should be retrieved together to be complete, a user requesting only part of the data can be advised that the output may be misleading, or the remainder of the data could be automatically retrieved and presented to the user as an addition to the requested output. A primary difficulty comes with identifying the data combinations. Personnel familiar with a single data base may be able to determine the meaningful combinations of data for that base, but, as multiple data bases are incorporated into the network and the user can retrieve and combine data from any of the system data bases simultaneously, the difficulty of identifying incomplete data combinations is increased significantly.

Another aspect of the same problem concerns data that, although unclassified, is for limited distribution. In determining what data are to be retrieved in response to a query, the NAVLIS system processes with successively more specific levels of identification. Facilities with data relevant to the query are identified, then data bases resident at that facility, records within the data bases, data elements within the records, and sub-elements of data elements. Restricted access can be recognized and enforced at any of these levels. For some users, access might be unequivocably denied. A dialogue could be initiated with other users to determine authorization. Other conditions might require notification of an individual responsible for granting or denying access based on the user and his requirements. The NAVLIS Pilot Model included an elementary provision for allowing access to certain data bases only through a three-character user or facility identification. In the Pilot Model access restriction was at the file level, i.e., a user was either permitted or denied access to an entire data base. It might be desirable, for example, to permit only a subset of the system users to retrieve from a certain personnel file. Restriction at a lower level, either the record or data element level, would also be required for an operational system. The NAVLIS system design incorporates two modules well-suited for providing more selective data privacy.

The File Selection Module (FSM) currently enables the system software to determine which data bases in the network can respond to a query by comparing the data elements involved in the query with the data elements contained in the system data bases. The module uses a support file, the File Content Directory, which contains the identification and location of each data base in the network associated with a list of the data elements in that data base. The file-level access verification is currently performed by this module, and the FSM is the logical point for enforcing data element restriction. The user's identification could be included in the comparison of the data elements in the query with those in the data base to determine whether the user was allowed access to those data elements in that data base. The File Content Directory could be expanded to indicate not only the data element content of each data base, but also, for those data elements with limited access, the user's who were permitted to retrieve those data elements. This capability would provide for allowing a user access to a personnel file for all but data concerning, for example, an individual's past and current earnings. In this way any unauthorized retrieval attempts can be detected very early in the processing cycle before any remote site transmission has occurred and prior to opening any of the system data bases.

It would be also useful to be able to restrict access on the basis of record type and certain data values rather than by data types. This type of restriction is difficult to accommodate without accessing the data base, but can be effected prior to making the data available to the user. The ADS Interface Module provides the logical point in the processing cycle to enforce both record and value restrictions. After retrieval of a record from the data base, the record type can be compared to any user ID restrictions for that data base before retrieving data values from the record. When no restrictions apply or when the user is permitted access to that record type, processing continues normally. If the user is not allowed access to that record type, a warning may be issued, processing may be terminated, or the record may be ignored and the next hit record accessed for processing. Summary type information might be available to a large class of users, but access to detail records would be limited.

Restricting access on the basis of the values of certain data elements could also be incorporated and could be performed by the ADS Interface Module. In the personnel data base example, a user may be allowed access to records for all personnel except those with certain job classifications. When such restrictions are indicated, the ADS Interface Module can test the value of restricting data elements in conjunction with the user ID before retrieving the requested data elements. Note that the data elements requested need not include the data element(s) on which access is restricted to prevent retrieval.

8.2 RECOMMENDATIONS

The following recommendations are presented without explanation of the area to which each recommendation applies, acknowledging that a reader needs an understanding of each area to fully appreciate the potential of a given recommendation.

(a) The NAVLIS Executive and Telecommunications Subsystem should be examined for possible utilization in other logistics systems such as WWMICS and NAICOMMIS.

(b) Consideration should be given to utilizing NAVLIS technology to network the Navy Data Processing Service Centers.

(c) Consideration should be given to implementing the NAVLIS Executive and Telecommunications Subsystems at DPSCLANT to serve the TYCOMS in the Norfolk area and other outside DPSCLANT users through on-line applications.

(d) Differences in perception of NAVLIS utility between LANTFLT and PACFLT should be pursued to determine whether the concept does indeed have a useful role in support of the fleet.

(e) Attention should be given to the technology developed in the NAVLIS project as it relates to on-going ONR sponsored programs at the university level.

ACKNOWLEDGMENTS

The Navy Logistics Information Sharing (NAVLIS) Project Office, DTNSRDC, Code 1806, acknowledges the following individuals for their efforts in support of NAVLIS Project development and documentation:

CDR Michael Bishop, NAVLIS Technical Manager for the first year of the project; Mr. Howard L. Mangan, NAVLIS Pilot Development Task Leader, for his direction of the development of the NAVLIS Executive Software and its documentation as Appendix A of this report; the technical programming and analytical staff consisting of Mr. David P. Carder, Ms. Bonnie H. Vincent, Mr. Thomas Martin, Ms. Joan Sullivan, and Mr. Jack McNichols (NAVCOSSACT) for their professionalism and dedication to the realization of project milestones and objectives; Mr. Ben A. Wallis for his system design contributions in the areas of on-line system development, data retrieval, and data file structuring; Mr. Charles Rogers (NAVCOSSACT) for his assistance to the project office in establishing realistic schedules for project planning, and his staffs for their excellent effort in developing the NAVLIS Telecommunications Network (NTS); finally, all the contacts and their staffs at the various organizations involved with the NAVLIS development.

Philip S. Braxton
NAVLIS Technical Manager

APPENDIX A

NAVLIS EXECUTIVE SUBSYSTEM SOFTWARE DOCUMENTATION

A.1 INTRODUCTION

This appendix represents the software documentation for those developments that are unique to a NAVLIS-type system. As mentioned in the body of the final report to which this is appended, the data management functions were absorbed into the NAVLIS software from an existing DMS (SHARP). Additional module development was required for the functions of system control, file selection, hit resolution, synonym resolution, and interface with the data bases against which retrieval would be effected. These functions represent (a) the differences between NAVLIS and other developments in data retrieval, and (b) the technical accomplishments necessary to demonstrate system feasibility and to implement the system at the Pilot Model sites. This appendix describes only those modules unique to NAVLIS, which comprise the Executive Control Subsystem. Those functions typically associated with data management systems, i.e., query interpretation, report generation, etc., are not documented herein. The format is as follows for each module described:

- Title and acronym(s) of module
- Purpose: Statement of the purpose(s) of the module.
- Inputs: A description of the inputs to the module and their source.
- Outputs: A description of the outputs from the module and their destination.
- Record Formats: Detailed record layouts for input, output, and records generated for internal use.
- Storage Requirements: Statement indicating the size of the module.

• Processing Description: Narrative statement of module processing coinciding with the program flow chart.

• Program Flowchart: A graphic statement of module processing.

Section 5.0 in the final report provides a brief description of the sequence of module execution that will assist the reader in putting the descriptions of individual modules into the content of system execution.

Sections A.9 and A.10 are modules unique to the requirement to interface with IBM's IMS data management system at NMCSA. These modules represent the interface necessary to monitor and capture IMS transactions against the data base for use in updating the Inverted Master Files for that data base.

A.2 TITLE: EXECUTIVE CONTROL MODULE (ECM)

A.2.1 PURPOSE: To control processing of queries and report definitions for displaying the results of the retrieval to the user. ECM also handles communications and error recovery during these processes.

A.2.2 INPUTS

The main inputs to ECM are the Module Communications Region (MCR) and an I/O buffer area (NTS-BUFFER) found in the Linkage Section between the NAVLIS Telecommunications Subsystem (NTS)* and ECM. This section contains the indicators and data from NTS necessary for accurate flow and processing. The Constraint File (CONSTR-FILE) is read by ECM following creation by the Query Translator Module (QTM) and modification by the Screening Module (SCRM) and the File Selection Module (FSM). CONSTR-FILE contains the translated constraints which the user has specified in his query for retrieval. The Edit File (EDIT-FILE) contains the reformatted records generated by the Report Generator Module (RGM) for displaying the results of the user's query in the desired format.

---

* The NAVLIS Telecommunications Subsystem (NTS) is documented under separate cover by NAVCOSSACT. See Reference 1.

A.2.3 OUTPUTS

The Linkage Section described above also acts as an output area for ECM in the communication between ECM and NTS. In addition, two output files are generated. The Scratch File (SCRATCH-FILE) is a collection of undeliverable records kept for later recovery. The Response File (RSVP-FILE) contains the records retrieved from the full data files in response to the user's query.

A.2.4 RECORD FORMATS: See Figures A1-A7

Module Communications Region

Data Field	Position	Picture
Function code	1-2	S99
Error code	3-4	S99
Request code	5-6	S99
Line advance	7-8	S99
Return code	9-10	S99
Message size	11-12	S99
Type code	13	X
Terminal ID	14-15	XX
Precedence	16	X
Station serial no.	17-20	9(4)
Content indicator	21-24	X(4)
Classification	25	X
Filler	26	X
Time of day	27-32	9(6)
Buffer size	33-34	S99
Module name	35-40	X(6)

Constraint File (Asterisk Record)

Data Field	Position	Picture
Asterisk	1	X
Site ID	2-7	X(6)
File ID	8-10	XXX
Filler	11-80	X(70)

Edit File

Data Field	Position	Picture
Carriage control	1	X
Rest of information	2-72	X(71)
Filler	73-80	X(8)

Response File

Data Field	Position	Picture
Filler	1-16	X(16)
Character count	17-18	99
Variable data	19-114	X(96)

Site Status Table

Data Field	Position	Picture		
Status field	1	X		
Site	2-7	X(6)		occurs 5 times
Query ID	8-10	XXX		
Question no.	11-13	999	occurs 5 times	
No. of hits	14-18	9(5)		

A.2.5 STORAGE REQUIREMENTS

Program (less library modules) occupies 24,000 bytes on UNIVAC Series 70/45.

Section & Page 101.
Subsection
Date Documented
Change Notice #

Fold back at dotted line.

CARD OR RECORD LAYOUT – DOUBLE

SYSTEM ID: ☐ New ☐ Revision

RECORD TITLE: SITE / FILE ID RECORD (*)
FILE ID: QUERY CONSTRAINT FILE
RECORD LENGTH: 80 MINIMUM ☐ FIXED ☑ MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR 1 RECORD/BLOCK

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON–CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7–TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

ASTERISK | SITE ID | FILE ID | FILE TYPE | FILLER

(1) (2) (3) (4)

(1) "*"
(2) ALPHANUMERIC
(3) ALPHANUMERIC
(4) "R" OR "S"

RECORD TITLE: MERGE RECORD (A)
FILE ID: QUERY CONSTRAINT FILE
RECORD LENGTH: 80 MINIMUM ☐ FIXED ☑ MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR 1 RECORD/BLOCK

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON–CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7–TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

QUERY ID | QUEST NO | FILLER | TYPE CODE | SET NO | FILLER

(1) (2) (3) (4)

(1) ALPHANUMERIC
(2) NUMERIC
(3) "A"
(4) 00 OR 01

Figure A-1

Section & Page 101.
Subsection
Date Documented
Change Notice #

Fold back at dotted line.

CARD OR RECORD LAYOUT – DOUBLE

SYSTEM ID: ☐ New ☐ Revision

RECORD TITLE: REPORT SELECTION RECORD (B)
FILE ID: QUERY CONSTRAINT FILE

RECORD LENGTH: 80 MINIMUM ☐ FIXED ☑ MAX.

WORDS ☐ BYTES ☐ CHARACTERS ☑

BLOCKING FACTOR 1 RECORD / BLOCK

MEDIUM: CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON–CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7–TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

QUERY ID	QUEST NO	FILLER	TYPE CODE	SET NO	FILLER	LITERAL	FILLER	FACILITY ID	FILLER	REPORT ID	FILLER
(1)	(2)		(3)	(4)		(5)		(6)		(7)	

(1) ALPHANUMERIC
(2) NUMERIC
(3) "B"
(4) NUMERIC
(5) "REPORT"
(6) ALPHANUMERIC
(7) ALPHANUMERIC

RECORD TITLE: COMPUTATION RECORD (C)
FILE ID: QUERY CONSTRAINT FILE

RECORD LENGTH: 80 MINIMUM ☐ FIXED ☑ MAX.

WORDS ☐ BYTES ☐ CHARACTERS ☑

BLOCKING FACTOR 1 RECORD / BLOCK

MEDIUM: CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON– CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7–TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

QUERY ID	QUEST NO	FILLER	CONTROL IN	TYPE CODE	SET NO	COMP CODE	LIRC-1	FILLER	ACRONYM-1	FILLER	FORM 1	DECIMAL 1	FILLER	LIRC-2	FILLER	ACRONYM-2	FILLER	FORM 2	DECIMAL 2	FILLER
(1)	(2)		(3)	(4)	(5)	(6)	(7)		(8)		(9)	(10)								

(1) ALPHANUMERIC
(2) NUMERIC
(3) "K", "N" OR "P"
(4) "C"
(5) NUMERIC
(6) "A" ··· "N"
(7) NUMERIC
(8) ALPHANUMERIC
(9) "I" "R" "N" "A" "L" OR SPACE
(10) NUMERIC OR SPACE

Figure A-2

Section & Page 101.
Subsection
Date Documented
Change Notice #

Fold back at dotted line.

CARD OR RECORD LAYOUT – DOUBLE

SYSTEM ID: ☐ New ☐ Revision

RECORD TITLE: SORT RECORD (D)
FILE ID: QUERY CONSTRAINT FILE
RECORD LENGTH: 80 MINIMUM ☐ FIXED ☑ MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR 1 RECORD/BLOCK

MEDIUM: CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON–CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7–TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Columns	1–3	4–6	7–8	9	10–11	12	13–16	17–26	27–30	31–82	83–100
Field	QUERY ID	QUEST NO	FILLER	TYPE CODE	SET NO	HIGH/LOW	LIRC	PREFIX/PARTIAL	CHAR COUNT	FILLER	(1) ALPHANUMERIC (2) NUMERIC (3) "D" (4) NUMERIC (5) "H" "L" OR SPACE (6) NUMERIC * (7) (NO) (≠NO), OR (≠NO, NO) (8) NUMERIC
	(1)	(2)		(3)	(4)	(5)	(6)	(7)	(8)		

* NO = 000 ··· 999

RECORD TITLE: HEADING RECORD (H)
FILE ID: QUERY CONSTRAINT FILE
RECORD LENGTH: 80 MINIMUM ☐ FIXED ☑ MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR 1 RECORD/BLOCK

MEDIUM: CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON–CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7–TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Columns	1–3	4–6	7–8	9	10–11	12	13–72	73–80	81–100
Field	QUERY ID	QUEST NO	FILLER	TYPE CODE	SET NO	FILLER	VARIABLE HEADING	FILLER	(1) ALPHANUMERIC (2) NUMERIC (3) "H" (4) NUMERIC (5) ALPHANUMERIC
	(1)	(2)		(3)	(4)		(5)		

Figure A-3

Section & Page 101.
Subsection
Date Documented
Change Notice #

Fold back at dotted line.

CARD OR RECORD LAYOUT - DOUBLE

SYSTEM ID: New ☐ Revision ☐

RECORD TITLE: SELECTION RECORD (J)
FILE ID: QUERY CONSTRAINT FILE
RECORD LENGTH: 80 MINIMUM ☐ FIXED ☑ MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR 1 RECORD/BLOCK

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Columns	Field
1–4	QUERY ID (1)
5–6	QUEST NO (2)
7–8	FILLER
9	TYPE CODE (3)
10–11	SET NO (4)
12	FILLER
13–16	LIRC (5)
17–26	FILLER
27	PARALLEL (6)
28–81	FILLER

(1) ALPHANUMERIC
(2) NUMERIC
(3) "J"
(4) NUMERIC
(5) NUMERIC
(6) "P" OR SPACE

RECORD TITLE: CONSTRAINT RECORD (L-Z)
FILE ID: QUERY CONSTRAINT FILE
RECORD LENGTH: 80 MINIMUM ☐ FIXED ☑ MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR 1 RECORD/BLOCK

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Columns	Field
1–4	QUERY ID (1)
5–6	QUEST NO (2)
7–8	OP CODE (3)
9	TYPE CODE (4)
10–11	SET NO (5)
12	MIDDLE (6)
13	LINK (7)
14	RULE (8)
15	SEARCH TYPE (9)
16–19	LIRC (10)
20–21	NO LINKS (11)
22–24	NO SETS (12)
25–26	INVERSION (13)
27	PARALLEL (14)
28	EFFICIENCY (15)
29–78	SEARCH VALUE (NON-RANGING SEARCH) / VALUE 1 (RANGING SEARCH) · VALUE 2 (RANGING SEARCH) (16)
79–80	FILLER

(1) ALPHANUMERIC
(2) NUMERIC
(3) ALPHABETIC
(4) "L" -- "Z"
(5) NUMERIC
(6) NOT USED
(7) NOT USED
(8) NOT USED
(9) "W" "X" "P", OR "S"
(10) NUMERIC
(11) NOT USED
(12) NOT USED
(13) NUMERIC
(14) "P" OR SPACE
(15) "X" OR SPACE
(16) ALPHANUMERIC

Figure A-4

Section & Page 101.

Subsection

Date Documented

Change Notice #

Fold back at dotted line.

CARD OR RECORD LAYOUT - DOUBLE

SYSTEM ID:

☐ New ☐ Revision

RECORD TITLE: PARALLEL LIRC RECORD (9)

FILE ID: QUERY CONSTRAINT FILE

RECORD LENGTH: 80 MINIMUM ☐ FIXED ☑ MAX.

WORDS ☐ BYTES ☐ CHARACTERS ☑

BLOCKING FACTOR 1 RECORD/BLOCK

MEDIUM: CARD ☐ TAPE ☐ DASD ☐

OTHER ☐

DENSITY

TRACKS: 7 ☐ 9 ☐

CARD STOCK

CHARACTER ENCODING:

HOL ☐ EXH ☐

BCD ☐ EBCDIC ☐ ASCII ☐

PACKED DECIMAL ☐

BINARY OR NON-CHARACTER INFORMATION ☐

OTHER ☐

PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:

TRAILER:

COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

QUERY ID	QUEST NO	FILLER	TYPE CODE	SET NO	FILLER	LIRC	FILLER	PARALLEL LIRC 1	FILLER	PARALLEL LIRC 2	FILLER	PARALLEL LIRC 3	FILLER	PARALLEL LIRC 4	FILLER	PARALLEL LIRC 5	FILLER	PARALLEL LIRC 6	FILLER	PARALLEL LIRC 7	FILLER	PARALLEL LIRC 8	FILLER	PARALLEL LIRC 9	FILLER	PARALLEL LIRC 10	FILLER
(1)	(2)		(3)	(4)		(5)		(6)																			

(1) ALPHANUMERIC
(2) NUMERIC
(3) "9"
(4) NUMERIC
(5) NUMERIC
(6) NUMERIC OR SPACES

RECORD TITLE:

FILE ID: SITE STATUS TABLE

RECORD LENGTH: 18 MINIMUM ☐ FIXED ☑ MAX.

WORDS ☐ BYTES ☐ CHARACTERS ☑

BLOCKING FACTOR 1 RECORD/BLOCK

MEDIUM: CARD ☐ TAPE ☐ DASD ☐

OTHER ☐

DENSITY

TRACKS: 7 ☐ 9 ☐

CARD STOCK

CHARACTER ENCODING:

HOL ☐ EXH ☐

BCD ☐ EBCDIC ☐ ASCII ☐

PACKED DECIMAL ☐

BINARY OR NON-CHARACTER INFORMATION ☐

OTHER ☐

PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:

TRAILER:

COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

STATUS	SITE	QUERY ID	QUEST NO	NUM OF HITS
(1)	(2)	(3)	(4)	(4)

(1) "*" OR SPACE
(2) ALPHANUMERIC
(3) OCCURS 5 TIMES
(4) OCCURS 5 TIMES

Figure A-5

Fold back at dotted line.

CARD OR RECORD LAYOUT – DOUBLE

Section & Page 101.
Subsection
Date Documented
Change Notice #

SYSTEM ID:

☐ New ☐ Revision

RECORD TITLE:
FILE ID: MODULE COMMUNICATIONS REGION
RECORD LENGTH: 40 MINIMUM ☐ FIXED ☑ MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)		(13)	(14)	(15)
FUNC CODES	ERROR CODE	REQ CODE	LINE ADV	RET CODE	MESS SIZE	TYPE CODE	TERM ID	PRECEDENCE	STATION SERIAL NO	CONTENT IND	CLASS	FILLER	TIME OF DAY	BUF	MODULE NAME

(1) NUMERIC
(2) NUMERIC
(3) NUMERIC
(4) NUMERIC
(5) NUMERIC
(6) NUMERIC
(7) ALPHANUMERIC
(8) ALPHANUMERIC
(9) ALPHANUMERIC
(10) NUMERIC
(11) ALPHANUMERIC
(12) ALPHANUMERIC
(13) NUMERIC
(14) NUMERIC
(15) ALPHANUMERIC

RECORD TITLE:
FILE ID: EDIT FILE
RECORD LENGTH: 80 MINIMUM ☐ FIXED ☑ MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR 5 RECORDS/BLOCK

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

(1)			
CARRIAGE	REST OF INFO	FILLER	

(1) "0", "1", OR SPACE

Figure A-6

Section & Page 101.

Subsection

Date Documented

Change Notice #

Fold back at dotted line.

CARD OR RECORD LAYOUT – DOUBLE

SYSTEM ID: ☐ New ☐ Revision

RECORD TITLE:

FILE ID: RESPONSE FILE

RECORD LENGTH: 20 MINIMUM ☑ FIXED ☐ 114 MAX.

WORDS ☐ BYTES ☐ CHARACTERS ☑

BLOCKING FACTOR 5 RECORDS/BLOCK

MEDIUM:

CARD ☐ TAPE ☐ DASD ☐

OTHER ☐

DENSITY

TRACKS: 7 ☐ 9 ☐

CARD STOCK

CHARACTER ENCODING:

HOL ☐ EXH ☐

BCD ☐ EBCDIC ☐ ASCII ☐

PACKED DECIMAL ☐

BINARY OR NON–CHARACTER INFORMATION ☐

OTHER ☐

PARITY: EVEN (7–TRACK BCD) ☐ ODD ☐

HEADER:

TRAILER:

COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Positions	1–6	7	8	9	10–11	12	13–16	17–18	19–100
Field	QUERY ID	QUEST NO	FILLER	NO PRINT	TYPE CODE	SET NO	SORT IND	LIRC	CHAR CNT	DATA VALUE

(1) (2) (3) (4) (5) (6) (7) (8) (9)

RECORD TITLE: CONTINUATION OF RESPONSE FILE

FILE ID:

RECORD LENGTH: MINIMUM ☐ FIXED ☐ MAX.

WORDS ☐ BYTES ☐ CHARACTERS ☐

BLOCKING FACTOR

MEDIUM:

CARD ☐ TAPE ☐ DASD ☐

OTHER ☐

DENSITY

TRACKS: 7 ☐ 9 ☐

CARD STOCK

CHARACTER ENCODING:

HOL ☐ EXH ☐

BCD ☐ EBCDIC ☐ ASCII ☐

PACKED DECIMAL ☐

BINARY OR NON– CHARACTER INFORMATION ☐

OTHER ☐

PARITY: EVEN (7–TRACK BCD) ☐ ODD ☐

HEADER:

TRAILER:

COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

(1) ALPHANUMERIC
(2) NUMERIC
(3) "*" OR SPACE
(4) "J"
(5) NUMERIC
(6) "Y" OR SPACE
(7) NUMERIC
(8) "01" ... "96"
(9) OCCURS 1 TO 96 TIMES

Figure A-7

A.2.6 PROCESSING DESCRIPTION: (Refer to ECM Flowchart, page 24)

The Executive Control Module (ECM) is called by the NAVLIS Telecommunications Subsystem (NTS) and calls the modules which comprise the retrieval subsystem. Processing in ECM is dependent on three indicators which control the flow of operation. These indicators are the NTS function code (MCRFNC), the ECM return code (MCRRET), and a relocator switch (RELOCATOR). The MCRFNC switch is set by NTS to inform ECM of the function to perform. The MCRRET switch is set by ECM to indicate an immediate return to ECM by NTS and subsequent transfer to that part of the program to be executed next. The RELOCATOR switch is an internal variable within ECM which acts as a secondary transfer code for a particular function code.

NTS enters ECM with the Module Communication Region (MCR) and the NTS-BUFFER. These variables enables NTS and ECM to communicate during execution. These calling parameters are displayed upon entry for debugging purposes.

Before checking MCRFNC to determine what function to perform, ECM must first check the message indicator (MESS-IND). ECM is responsible for carriage return and line advance when messages are displayed to the user. MESS-IND greater than zero indicates that a message has been put out and that ECM should return to NTS for carriage return and the proper line advance. MESS-IND = 1 indicates a single line advance. MESS-IND = 2 indicates no line advance. MESS-IND = 0 indicates the task has been performed or does not need to be performed, which leads to the GO TO --- DEPENDING ON MCRFNC. The following table lists the transfer points and functions indicated by the various codes:

MCRFNC	PARAGRAPH NAME	FUNCTION
1	INITIAL	Initialization
2	INITIAL	Restart
3	NO-OP	No function to perform
4	1ST-MID-BLK	Intermediate data block
5	FINAL-BLK	Final data block
6	1ST-MID-UND	Intermediate undeliverable
7	FINAL-UND	Final undeliverable
8	CONS-REQ	Console message request
9	TIME	Time of day

(Continued)

MCRFNC

10	NO-DATA	No data in the buffer
11	TERM-REQ	Request for termination
12	LINE-REACT	Line reactivated
13	IMMEDIATE-RETURN	Provide next data block

A function code of 0 is an error condition resulting in display of a message to the user and eventual termination of the run.

The INITIAL paragraph sets the various switches and indicators to their initial value, returning to NTS with "no request". The RELOCATOR switch is set to 11 (see RELOCATOR function summary).

NO-OP is executed when MCRFNC = 3. In this case, control has been given to NTS by ECM with the return code set. NTS has returned to ECM with no function to perform. Transfer is made to that part of ECM to be executed next via a GO TO --- DEPENDING ON MCRRET. Each of the paragraphs will be explained as they are encountered in the flow.

1ST-MID-BLK is the paragraph executed when there is a first or intermediate data block in the buffer from NTS. If this intermediate block contains the counter record from a remote site, the record is stored and ECM returns to NTS for the next data block. If the error code is set, control goes to the error routine. If ECM receives a function code of 4 during the log-in procedure, return is made to NTS with a request to purge the current data block. If none of the above are true, the data block is assumed to be a response record from a remote site. The record is written on the response file and ECM returns to NTS with a request for the next data block.

FINAL-BLK is executed when the buffer contains the final or only data block of a message. This is the most common function code.
A second level GO TO--- DEPENDING ON is used for transfer to the proper area of processing. This secondary transfer code is the RELOCATOR switch mentioned earlier. The following table lists the transfer points and function indicated by the RELOCATOR switch.

RELOCATOR	PARAGRAPH NAME	FUNCTION
1	FAC-CK	Validate facility id
2	USER-CK	Validate user id
3	PASSWRD-CK	Validate password
4	TRANSLA	Translate query
5	HIT-SELECTION	Disposition of hits
6	ANOTHER-QUERY	Prepare for next query
7	FOURTEENTH	Query option
8	OPTION	Query, repdef, or end option
9	NEXT-PARA	Hit resolution
10	OPEN-UP	Open inverted files
11	GET-FIRST	Prepare for facility id entry
12	ANOTHER-QUERY	Prepare for next query
13	GET-FIRST	Prepare for facility id entry
14	CALL-REPDEF	Report definition
15	OPT	Query, repdef, or end option
16	CALL-FSM	File selection

The 1ST-MID-UND paragraph handles the first or intermediate blocks which are undeliverable. If the last program executed was RGM or FSM, the undeliverable data block in the buffer is purged. MOD-IND=RGM indicates a report undeliverable to the user. MOD-IND=FSM indicates a constraint file undeliverable to the remote site. Neither of these need be saved because they may be obtained elsewhere. All other undeliverable data blocks, such as messages to the user, are saved on the scratch file for retransmission.

FINAL-UND writes the last undeliverable data block on the scratch file. If this is the only data block, the scratch file is first opened, then written on and closed. If the error indicator is set, control is transferred to ERR-CK for proper return to NTS.

CONS-REQ is the paragraph executed when a console request has been issued by NTS indicating there is something to be entered at the console. ECM returns to NTS with a no-request to allow the function to be performed.

TIME obtains the time of day from NTS and passes it to the user via NTS. Note that the return code is set for immediate return to ECM after the message is given to the user. This is for the carriage return and line advance.

NO-DATA indicates there is no data in the buffer. In this case, the error indicator is checked and control is transferred to the error paragraph for proper handling. If the error indicator is not set, the user is advised that unrecoverable errors have occurred and the run is terminated.

TERM-REQ handles a request for termination by NTS. ECM returns to NTS with a request code of 9 after performing the necessary housekeeping functions.

Control is passed to the LINE-REACT paragraph when a line is reopened after being down. If RGM was the last module executed before the line went down, the EDIT-FILE is closed and reopened for display to the user. If the last module executed was FSM, the CONSTR-FILE is reread and transmitted to the proper site for retrieval. If neither RGM nor FSM was executed last, the records on the scratch file are retransmitted to NTS for continued processing.

The IMMEDIATE-RETURN paragraph is executed whenever ECM requests that NTS return to ECM immediately after completing the current request. Transfer is made to that part of ECM to be executed next via a GO TO--- DEPENDING ON MCRRET. If this return code has not been set, the user is notified and the run is terminated.

When a user dials into the NAVLIS system, NTS calls ECM with a function code of 1 indicating initialization. ECM then sets the various switches and returns to NTS with RELOCATOR = 11. The user then logs into the system by entering "NLIS". NTS passes this message to ECM via the NTS-BUFFER with a function code of 5 indicating that the buffer contains a final or only data block. Since the RELOCATOR switch had been set by ECM before the last return to NTS, control is passed, first to the FINAL-BLK paragraph, and then to GET-FIRST where ECM asks the user to enter his facility id. The RELOCATOR switch is again set to transfer control to FAC-CK for validation of the facility id. The user is given three attempts at a valid facility id. If he does not succeed, he is advised to get help, and the run is terminated. Entry and validation of the user id and password is similar. For security, however, the

password is blacked out before entry by overwriting the entry line with a series of literals which completely fills each character matrix. The RELOCATOR switch then transfers control to OPTION. The OPTION paragraph marks the end of the logon stage. The user is given the choice of QUERY, REPDEF, or END.

The REPDEF option enables the user to define report formats from the terminal. ECM calls the REPDEF module for each report line entered by the user. The calling parameters indicate whether ECM should return to NTS for the next line from the user (MCRREQ = 2), put out an error message, or end the procedure.

The END option simply terminates the run through a request code of 7. NTS then completes its housekeeping and disconnects the line.

The QUERY option permits the user to interrogate the desired data files and receive the results at the terminal. The first phase of this interrogation is the Query Translator Module which translates the English-like query entered by the user to a form which can be understood by the retrieval system. ECM calls QTM for each line of input. As in REPDEF, the calling parameters indicate whether ECM should return to NTS for the next line from the user, put out an error message, or end the procedure.

When the user's query has been translated, the user is advised of its acceptance. The return code is set for an immediate return (MCRRET = 12) and ECM returns to NTS. Note that the return code is set for transfer in this case rather than the RELOCATOR switch. This is because no response is required from the user before returning to ECM. Most returns prior to this required user input which was then validated or processed accordingly. Immediate return to ECM after a message has been displayed does not permit or depend on this input. Control is transferred to that paragraph referenced by the return code. After query acceptance, control passes to TWELFTH in which the Screening Module is called. This program screens the output of QTM for report validation, etc.

The File Selection Module is the next program executed in the retrieval system. The user is asked if he would like to enter file parameters to limit the files accessed. If he responds with "yes", he is

directed to enter his parameters in a prescribed format. The module communicates with the user through the use of ERROR-IND and INDICAT. Each value of these calling parameters indicates a different level of response, enabling FSM to put out the proper message to the user and process his response correctly.

When file selection is complete, all remote site constraint records are transmitted to the appropriate site for retrieval. Each site is identified on the file with an asterisk record. When one of those records is encountered, the next record is read. If this next record is not an asterisk record, the previous record is transmitted to the proper site. This process is continued until the next asterisk record is encountered. Currently, the second asterisk record signals central site processing since we are operating with only two sites. During the transmission process, control fluctuates between PROCESS-FSM-FILE and FIFTH.

ERROR-PARA, which was referenced earlier, puts out the error messages from the various modules to the user. The messages are passed to ECM via the calling parameter, ERREA. ECM then passes the message to NTS line by line in NTS-BUFFER. NTS displays each line to the user and returns to ECM for the next line, and so on. When all lines of the message have been put out, ECM checks the various switches for transfer to the proper area of processing.

When all the appropriate constraint records have been transmitted to the remote site, the constraint records for the central site are processed. This is the RCM function of ECM. The processing performed at the remote site is duplicated for the central site records. OPEN-UP begins this procedure. The central site file to be searched is identified. Synonym resolution is then performed. If no errors are encountered by the Synonym Resolution Module, the Primary Hit Resolution Module is called to select hit candidates for the user's query from the specified inverted file. There are several error conditions out of PHRM, ERROR-IND = 0 indicates there were no errors and primary hit resolution is complete. ERROR-IND = 1 tells ECM that PHRM wishes to display a message to the user and receive control again. ERROR-IND = 9 implies that there were no hits for any question at the central site.

Successful completion of primary hit resolution results in a call to the ADS module associated with the current inverted file. This module retrieves the requested data from the corresponding direct files and returns the records one by one to ECM via the NTS-BUFFER. ECM writes these records on the response file as they are received. When ADS returns to ECM for the last time, the NTS-BUFFER contains spaces, and the calling parameter, SITE-TABLE, contains a counter record which has the number of hits for each question from the current file. The hits are then accumulated for the central site. If no more files are to be processed at this site (ENDED = 1), the SITE-STATUS-TABLE is updated by transferring the accumulated hits and moving an asterisk to the status field to indicate completed processing. If more files are to be processed, control is transferred to OPEN-UP for file identification, etc.

UPDATE-SST checks the status field for each site. An asterisk in this field indicates completed processing. If all sites have returned their records, control is transferred to ADVISER for disposition by the user. If the remote site has not yet completed processing, ECM returns to NTS to wait for it.

When NTS returns to ECM with the response records from the remote site, the function code = 4 indicating a first or intermediate data block. The records are written on the response file as they are returned. The counter record for the remote site is passed with a function code of 4. ECM stores the counter record and returns to NTS for the final data block. The return code is set to 6 for return to SIXTH. MCRFNC = 5 and MCRMSZ = 0 indicate that the counter record was stored from the previous data block. The stored record is then moved to NTS-BUF to be handled as if the counter record had been passed in the NTS-BUF with a function code of 5, which is another possibility. The SITE-STATUS-TABLE is then updated for the remote site.

ADVISER computes the number of hits for each question at all sites. If none of the user's questions have hits, the user is notified and given the option of submitting another query. If he wishes to

do so, he is asked to input his query commands and control is transferred to TRANSLA via the RELOCATOR switch. If the user does not want to submit another query at this point, the run is terminated. If at least one of the user's questions has hits, he is advised of the number of hits and given the option of deleting or printing the results of each question. Those questions to be deleted are flagged in the DELETER table. If all questions are to be deleted, control is transferred to FOURTEENTH, where the user is asked if he wants to submit another query. If the results of one or more questions are to be printed, the Sort Module is called. This module sorts the data, eliminates deleted questions, and reformats for the Report Generation Module which is called next.

RGM develops the report in the format specified in the user's query and writes the result on EDIT-FILE. ECM then reads EDIT-FILE, converts the carriage control to NTS standards, and displays the report to the user line by line. Lines with all spaces require only a carriage return. Those with a carriage control of space require no advance (MCRADV = 0). Those with a carriage control of zero require a one line advance (MCRADV = -1 implies write after advancing one line). Those with a carriage return of 1 require a page advance (MCRADV = -5). When the entire report has been printed, the user is given the option of submitting another query. Processing then continues as described.

A.2.7 ECM/RCM CONSOLIDATION

The Remote Control Module (RCM) directs the processing of the retrieval modules at the remote site. In addition, it handles communications and error recovery. It is essentially a subset of ECM, performing synonym resolution through hit selection at the remote site. The two modules have therefore been combined into one for easier maintenance and control. There were differences in remote site processing which were incorporated in the combined version. These are noted below. See the flow chart for more detail.

A master version was developed containing UNIVAC and IBM code, identified by *U or *I in columns seven and eight. When a new load module is to be made for the IBM machine, the master version is first copied to another name. All coding beginning with *U is then edited out under Time Sharing Options (TSO). The remainder is then compiled to create a new load module. Note that the master version is not changed by this procedure.

An entry point was added to the new ECMRCM for RCM. This required an additional parameter, WKACPCB. The ADS Interface Module cannot access the IMS files directly due to certain IBM restrictions. NTS must, therefore, access the files, pass the results through WKACPCB to RCM which, in turn, passes it to the ADS Interface Module.

All module calls were modified for the RCM function of ECMRCM to accommodate IBM requirements of double quotes arount the module name. The quotes do not appear on the enclosed listing because the module was compiled with the quote option.

The remainder of the differences which were incorporated in the combined version of ECM and RCM are listed by function code in the chart below. Note that some of the functions which were performed by RCM are handled elsewhere by ECMRCM.

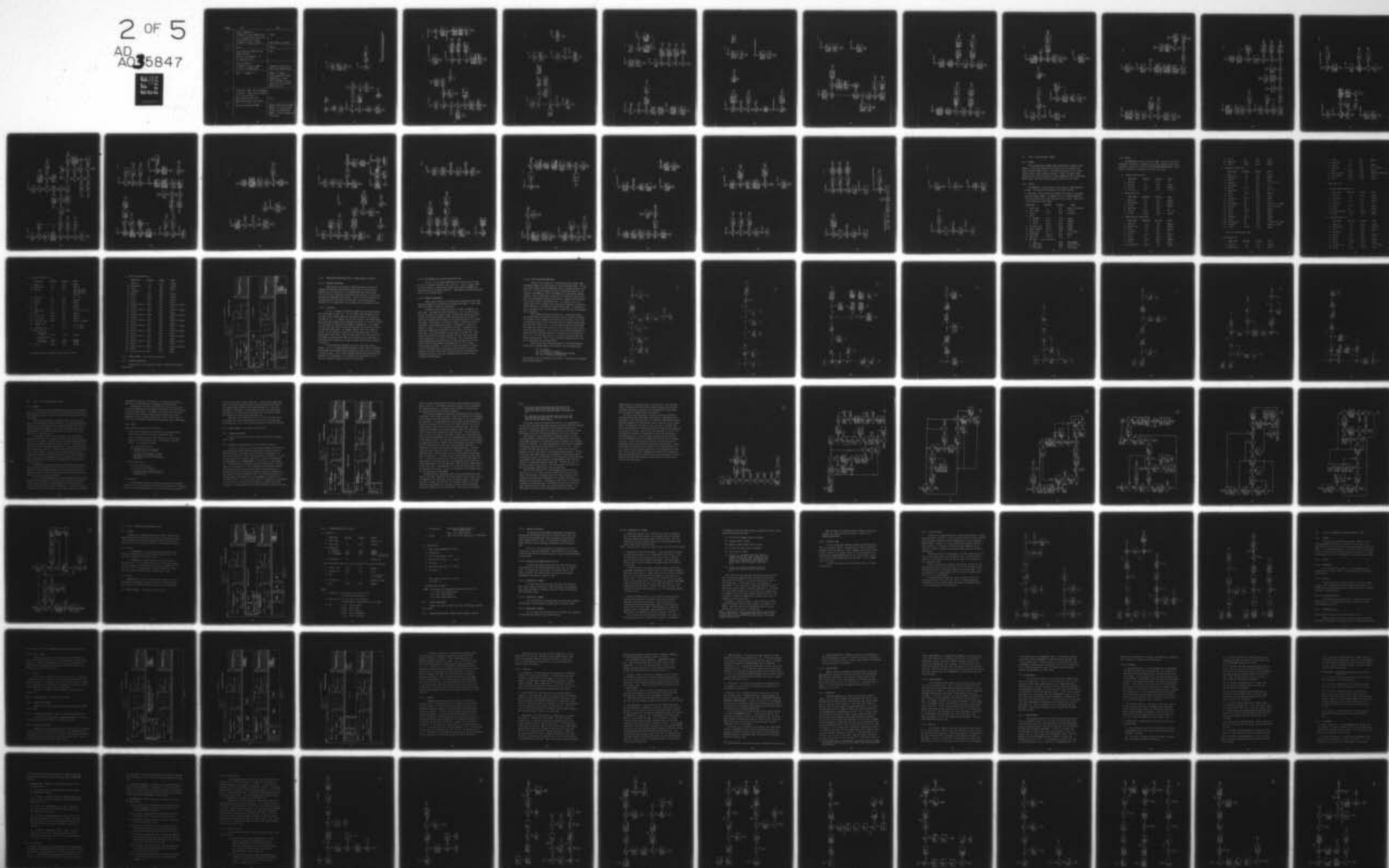
ECM/RCM Differences

MCRFNC	ECM	RCM
1/2	Sets RELOCATOR = 11 to control subsequent prompting messages	Checks CONSTR-FILE and SCRATCH-FILE and closes them if they're open. Clears the BREAK-DOWN array where hit counts are accumulated.
3	Control routed on MCRRET	Same
4	Checks for counter record and stores it if found. Passes control to error paragraph if error code is set. Purges current message if in log-in procedure. Opens RSVP-FILE in output mode if it is closed. Writes record to RSVP-FILE	Clears BREAK-DOWN array if its status is "in use". Opens CONSTR-FILE in output mode if it is closed. Writes record to CONSTR-FILE.
5	Routes control on STORE-ERR, RSVP-OPEN, MCRERR, or FAC-ID if set, but normally routes control on RELOCATOR (if 0 RELOCATOR 17) else on MCRRET (if 0 MCRRET 30) else generates an error message.	Opens CONSTR-FILE if not open, writes current block to CONSTR-FILE (if block ≠ spaces), and closes CONSTR-FILE. Hard coded to pass control to synonym resolution logic.
6	If MOD-IND = "FSM" or "RGM", purges the data Else opens SCRATCH-FILE in output mode if it is closed. Sets AT-LEAST-ONE = 1 Writes record to SCRATCH-FILE	Same but not as an else condition.

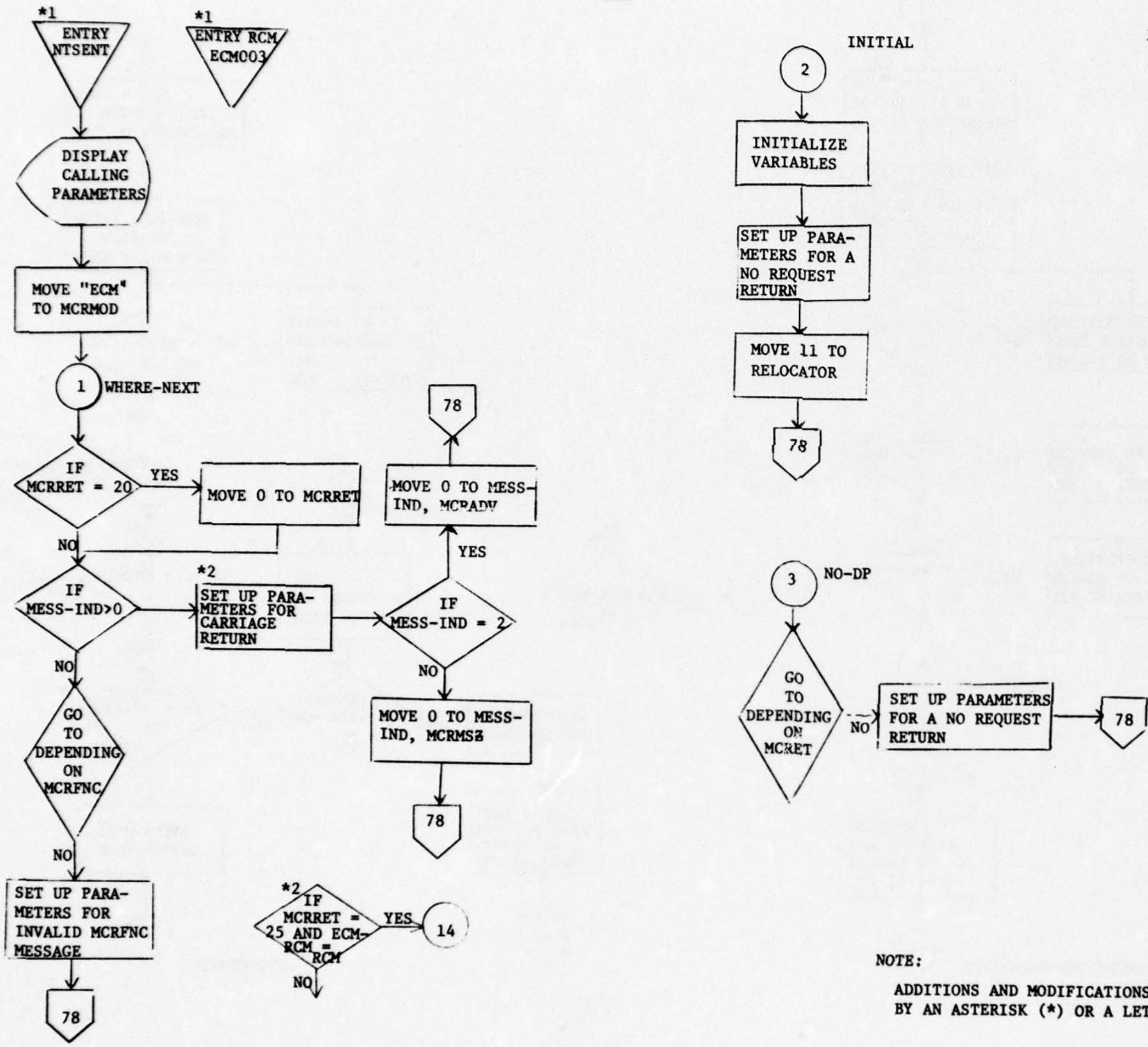
AD-A035 847 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 9/2
NAVY LOGISTICS INFORMATION SHARING (NAVLIS) PROJECT.(U)
JUN 76

UNCLASSIFIED DTNSRDC-76-0120 NL

2 OF 5
AD
A035847

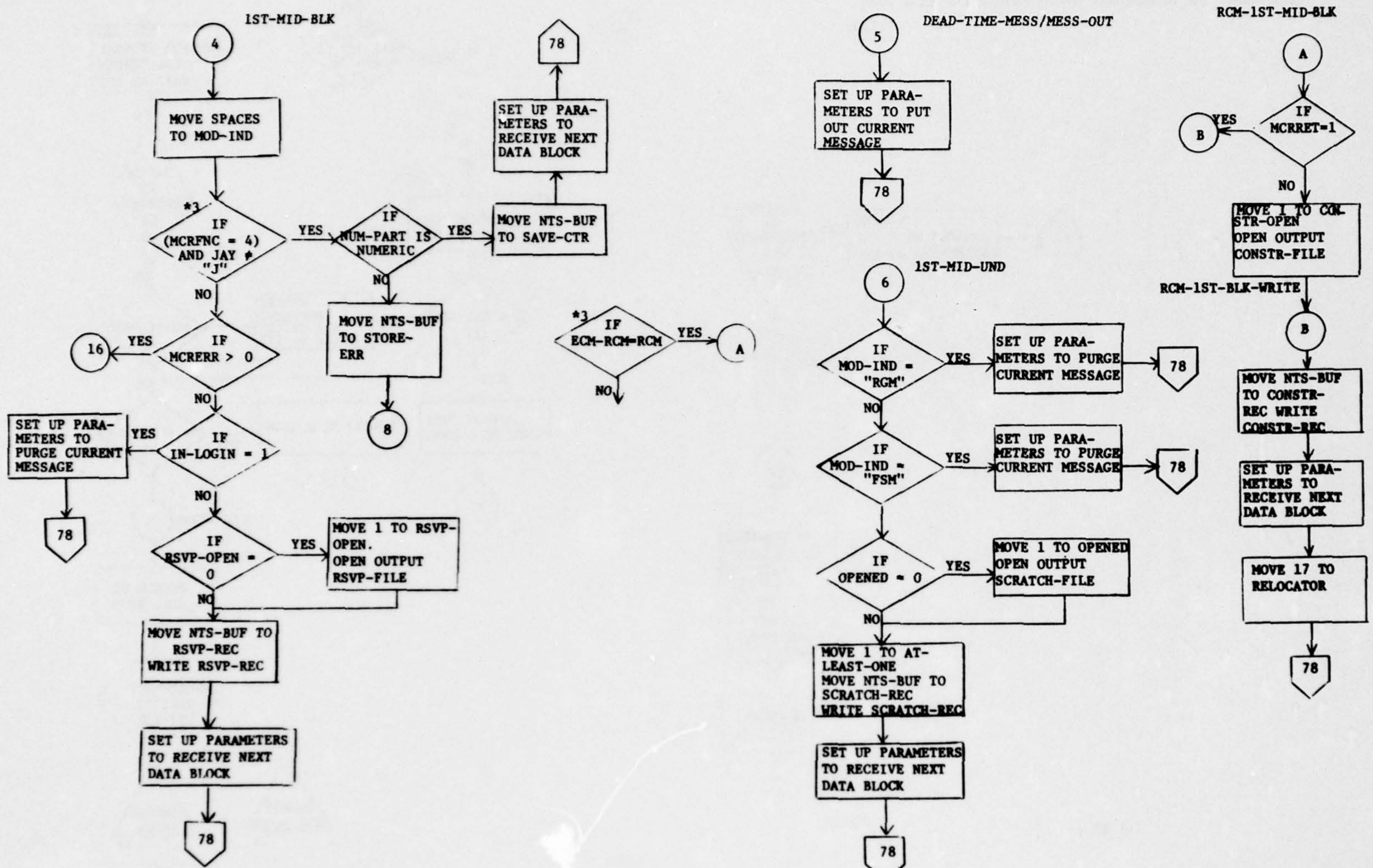


MCRFNC	ECM	RCM
7	Clears RELOCATOR = 0. If MCRMSZ > 0, opens SCRATCH-FILE if it is closed and writes record. Closes SCRATCH-FILE if open. If MCRERR > 0, goes to ERR-CK.	Same Moves MESS21 to NTS-BUF
8	Return	Return
9	Moves MCRTOD to TIMER array and TIMER to NTS-BUF Sets REQ = 5 and MCRRET = 20 for subsequent entry.	Same
10	Clears RELOCATOR = 0 If MCRERR > 0, goes to ERR-CK, else goes to INSERTED-PARA	 Same(Minor differences in ERR-CK and INSERTED-PARA)
11	MCRREQ = 9; MCRRET = 0	MCRREQ = 9; MCRRET = 0 Closes CONSTR-FILE and/or SCRATCH-FILE if open. Sets BREAK-DOWN switch to ensure array gets cleared before next use.
12	If MOD-IND = "RGM", go to FIFTEENTH. If MOD-IND = "FSM", go to SIXTEENTH. Exit if AT-LEAST-ONE = 0 Open SCRATCH-FILE for input Read and pass back first record.	 Same
13	Routing controlled by MCRRET	Routing controlled by MCRRET MCRRET = 7=> retrieve next record from SCRATCH-FILE MCRRET = 2 =>call ADSFACE for next record.

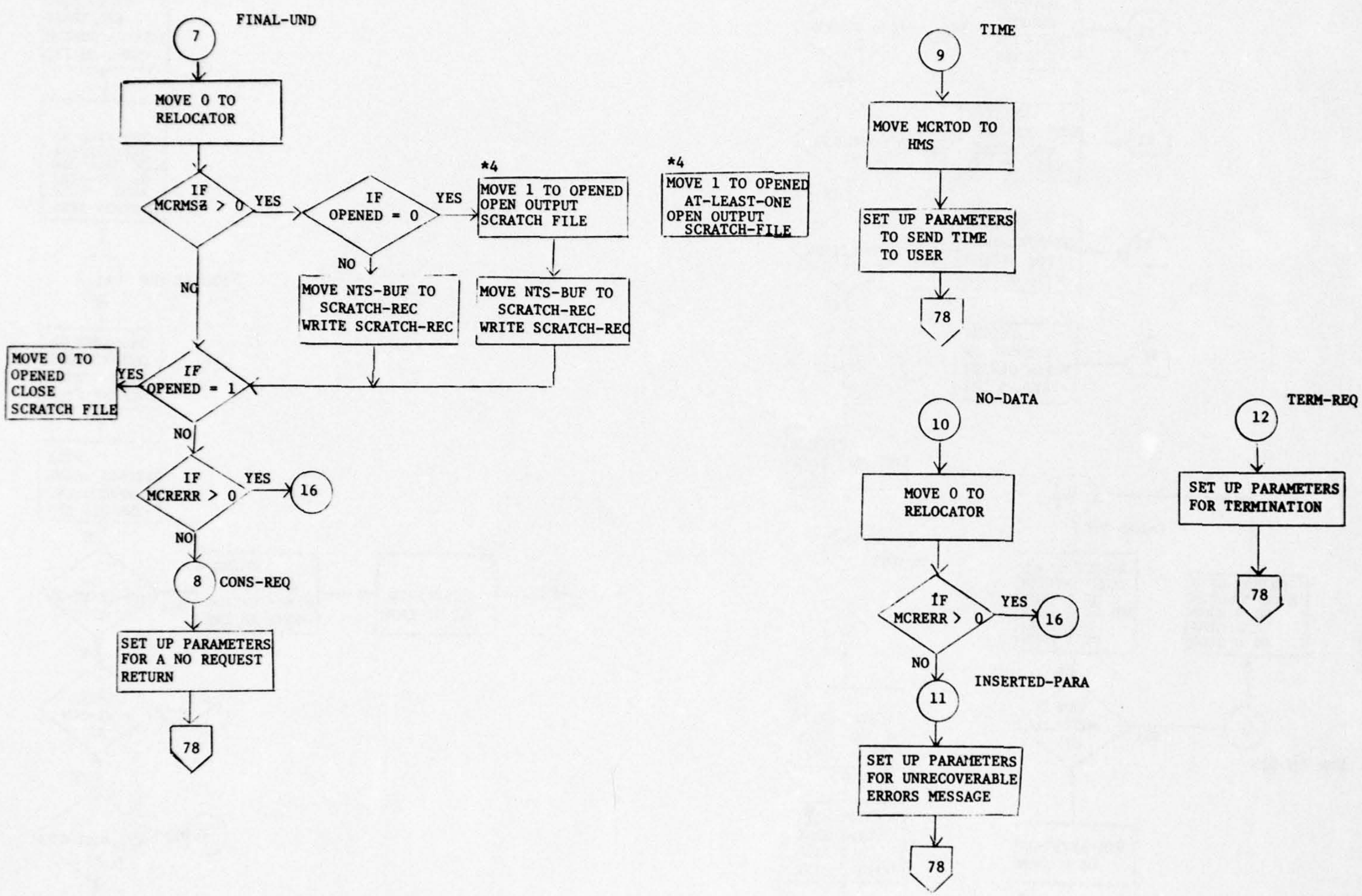


NOTE:

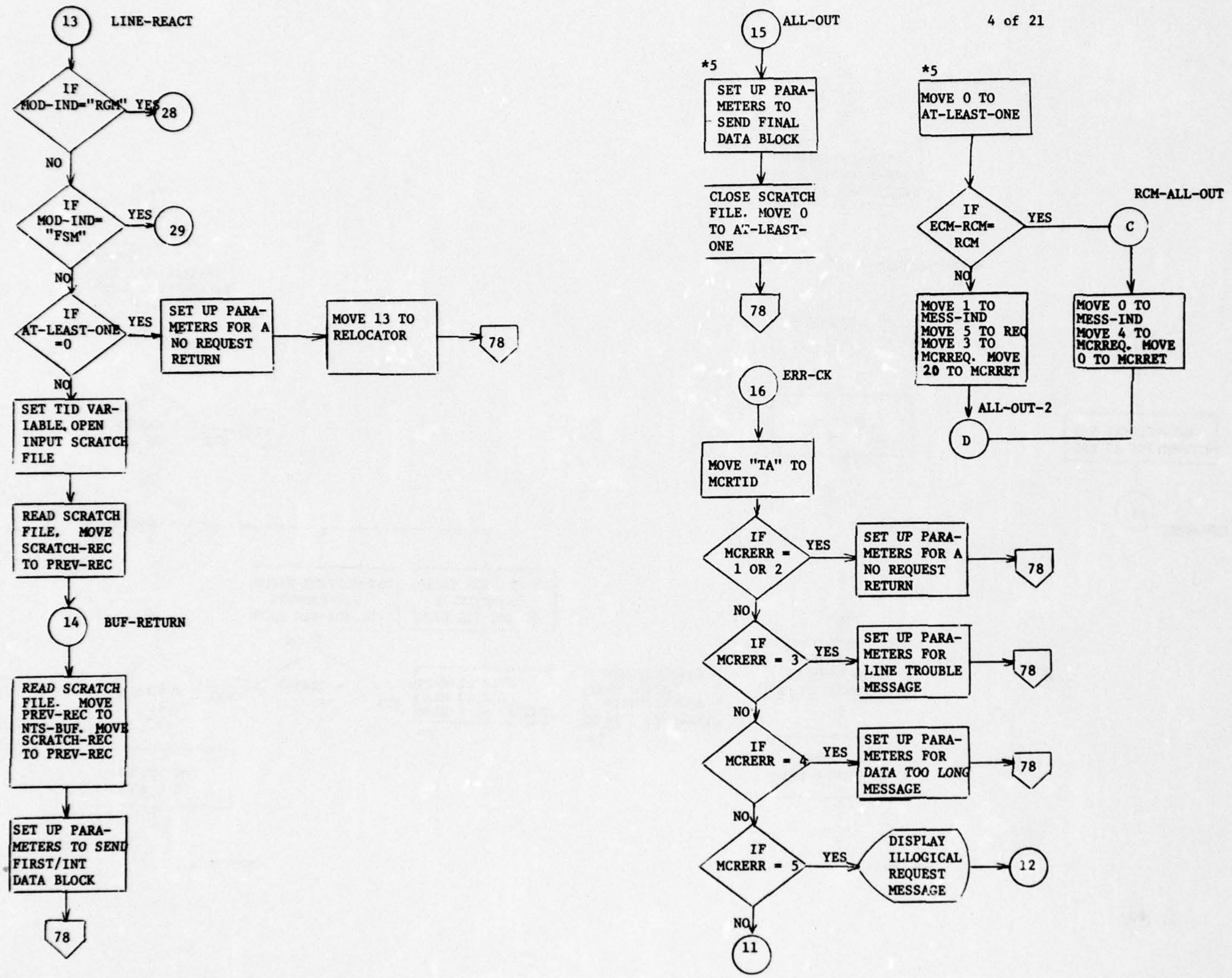
ADDITIONS AND MODIFICATIONS FOR ECMRCM ARE IDENTIFIED BY AN ASTERISK (*) OR A LETTER WITHIN A CIRCLE.

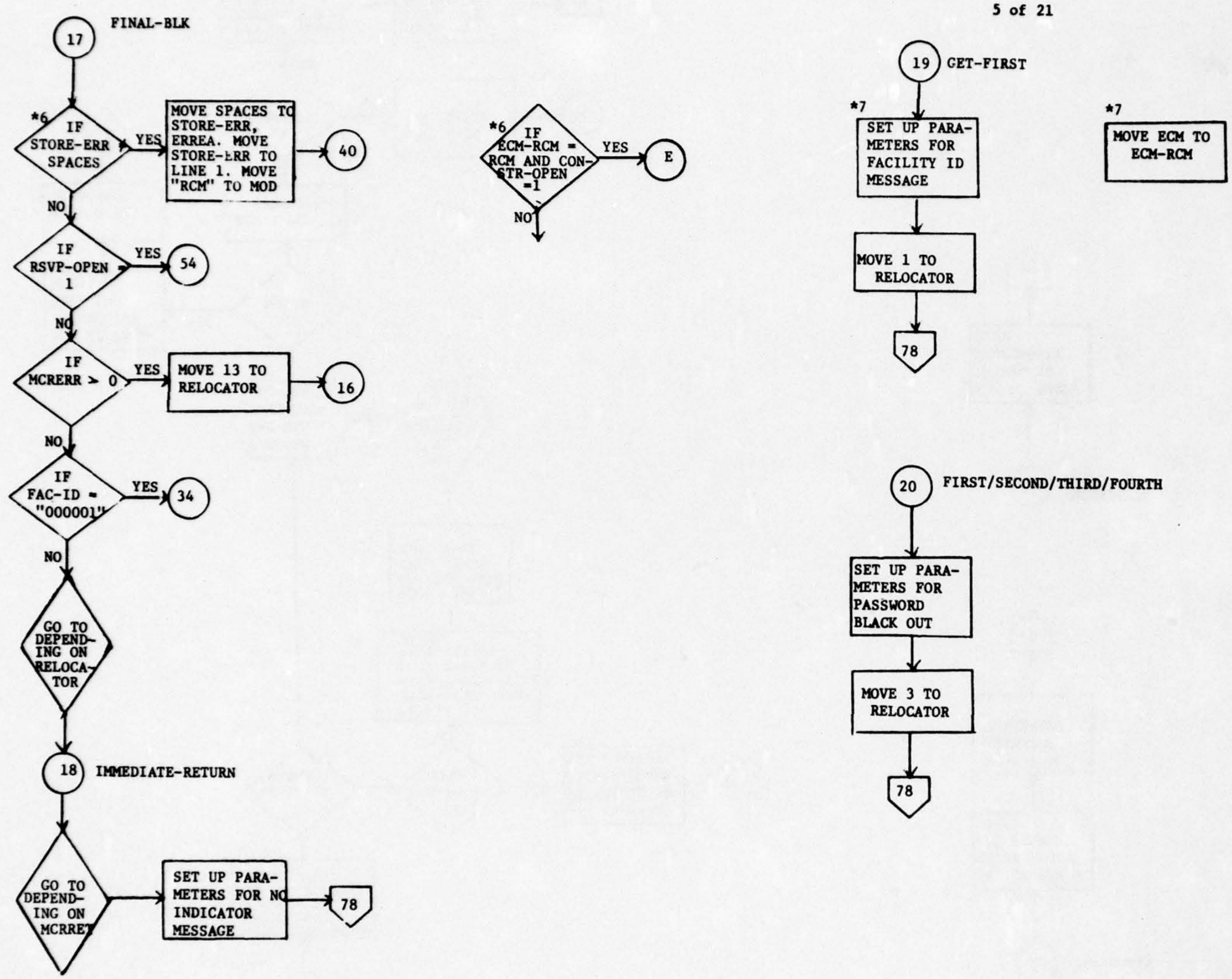


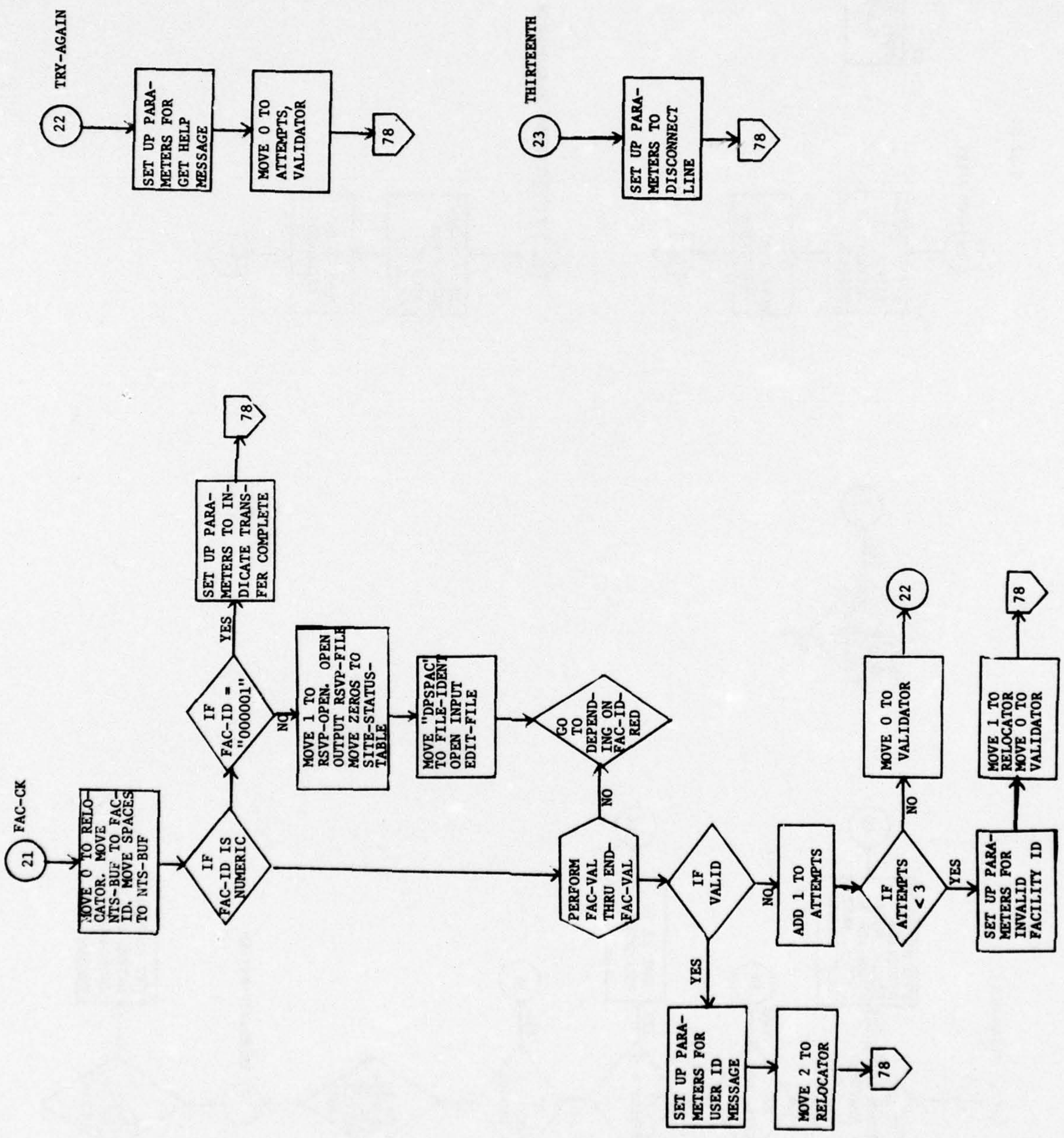
1ST-MID-BLK
4
MOVE SPACES TO MOD-IND
*3
IF (MCRFNC = 4) AND JAY ≠ "J"
YES
IF NUM-PART IS NUMERIC
YES
MOVE NTS-BUF TO SAVE-CTR
SET UP PARAMETERS TO RECEIVE NEXT DATA BLOCK
78
NO
MOVE NTS-BUF TO STORE-ERR
8
NO
IF MCRERR > 0
YES
16
NO
IF IN-LOGIN = 1
YES
SET UP PARAMETERS TO PURGE CURRENT MESSAGE
78
NO
IF RSVP-OPEN = 0
YES
MOVE 1 TO RSVP-OPEN. OPEN OUTPUT RSVP-FILE
NO
MOVE NTS-BUF TO RSVP-REC WRITE RSVP-REC
SET UP PARAMETERS TO RECEIVE NEXT DATA BLOCK
78
*3
IF ECM-RCM=RCM
YES
A
NO
DEAD-TIME-MESS/MESS-OUT
5
SET UP PARAMETERS TO PUT OUT CURRENT MESSAGE
78
1ST-MID-UND
6
IF MOD-IND = "RCM"
YES
SET UP PARAMETERS TO PURGE CURRENT MESSAGE
78
NO
IF MOD-IND = "FSM"
YES
SET UP PARAMETERS TO PURGE CURRENT MESSAGE
78
IF OPENED = 0
YES
MOVE 1 TO OPENED OPEN OUTPUT SCRATCH-FILE
NO
MOVE 1 TO AT-LEAST-ONE MOVE NTS-BUF TO SCRATCH-REC WRITE SCRATCH-REC
SET UP PARAMETERS TO RECEIVE NEXT DATA BLOCK
78
RCM-1ST-MID-BLK
A
IF MCRRET=1
YES
B
NO
MOVE 1 TO CONSTR-OPEN OPEN OUTPUT CONSTR-FILE
RCM-1ST-BLK-WRITE
B
MOVE NTS-BUF TO CONSTR-REC WRITE CONSTR-REC
SET UP PARAMETERS TO RECEIVE NEXT DATA BLOCK
MOVE 17 TO RELOCATOR
78



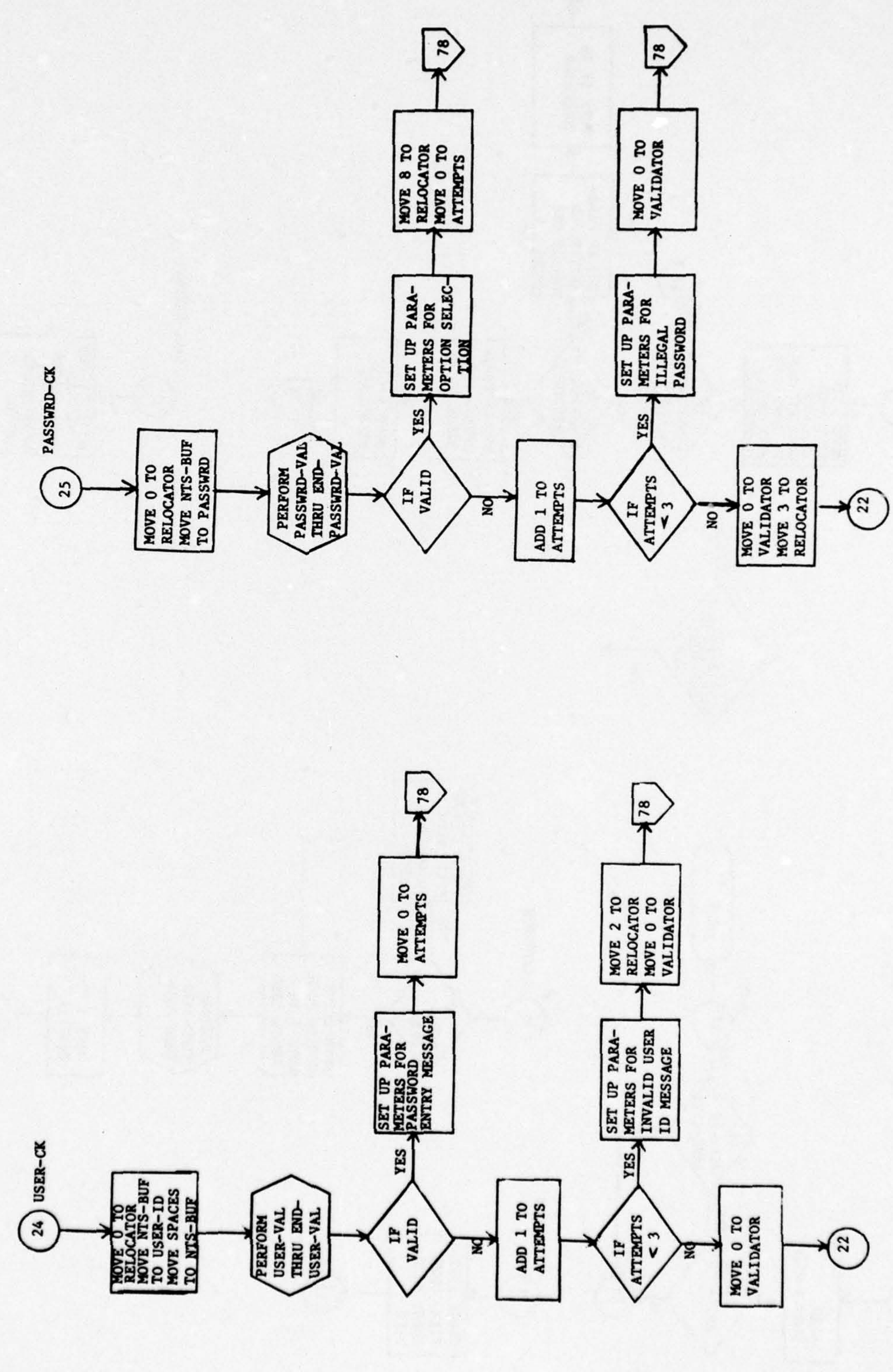
7
FINAL-UND
MOVE 0 TO RELOCATOR
IF MCRMSZ > 0
YES
NO
IF OPENED = 0
YES
NO
*4
MOVE 1 TO OPENED OPEN OUTPUT SCRATCH FILE
*4
MOVE 1 TO OPENED AT-LEAST-ONE OPEN OUTPUT SCRATCH-FILE
MOVE NTS-BUF TO SCRATCH-REC WRITE SCRATCH-REC
MOVE NTS-BUF TO SCRATCH-REC WRITE SCRATCH-REC
IF OPENED = 1
YES
MOVE 0 TO OPENED CLOSE SCRATCH FILE
NO
IF MCRERR > 0
YES
16
NO
8
CONS-REQ
SET UP PARAMETERS FOR A NO REQUEST RETURN
78
9
TIME
MOVE MCRTOD TO HMS
SET UP PARAMETERS TO SEND TIME TO USER
78
10
NO-DATA
MOVE 0 TO RELOCATOR
IF MCRERR > 0
YES
16
NO
11
INSERTED-PARA
SET UP PARAMETERS FOR UNRECOVERABLE ERRORS MESSAGE
78
12
TERM-REQ
SET UP PARAMETERS FOR TERMINATION
78

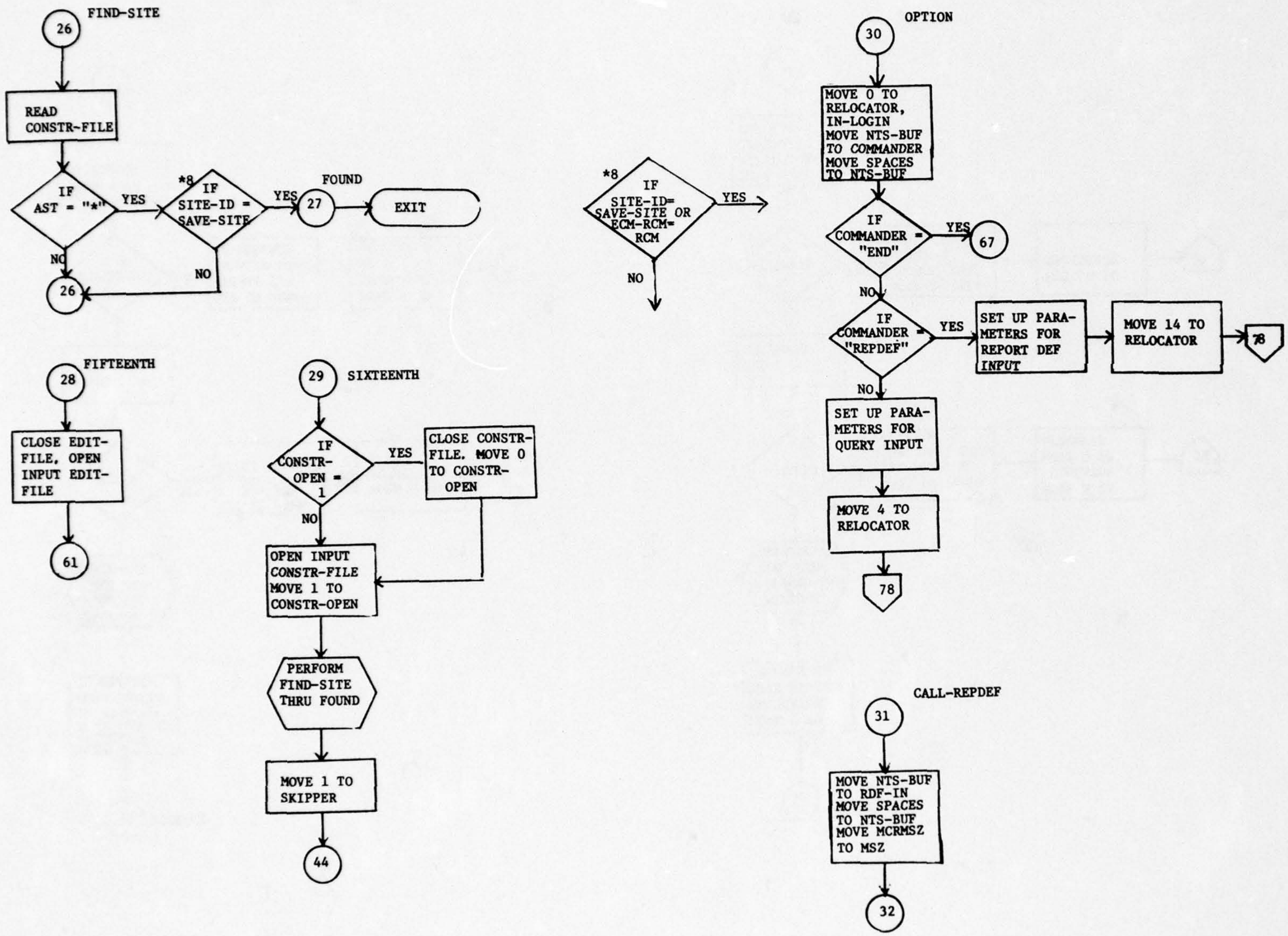

13 LINE-REACT
IF MOD-IND="RGM" YES 28
NO
IF MOD-IND="FSM" YES 29
NO
IF AT-LEAST-ONE =0 YES
SET UP PARAMETERS FOR A NO REQUEST RETURN
MOVE 13 TO RELOCATOR
78
NO
SET TID VARIABLE, OPEN INPUT SCRATCH FILE
READ SCRATCH FILE. MOVE SCRATCH-REC TO PREV-REC
14 BUF-RETURN
READ SCRATCH FILE. MOVE PREV-REC TO NTS-BUF. MOVE SCRATCH-REC TO PREV-REC
SET UP PARAMETERS TO SEND FIRST/INT DATA BLOCK
78
15 ALL-OUT
*5
SET UP PARAMETERS TO SEND FINAL DATA BLOCK
CLOSE SCRATCH FILE. MOVE 0 TO AT-LEAST-ONE
78
*5
MOVE 0 TO AT-LEAST-ONE
IF ECM-RCM=RCM YES
RCM-ALL-OUT
C
NO
MOVE 1 TO MESS-IND MOVE 5 TO REQ MOVE 3 TO MCRREQ. MOVE 20 TO MCRRET
MOVE 0 TO MESS-IND MOVE 4 TO MCRREQ. MOVE 0 TO MCRRET
ALL-OUT-2
D
16 ERR-CK
MOVE "TA" TO MCRTID
IF MCRERR = 1 OR 2 YES
SET UP PARAMETERS FOR A NO REQUEST RETURN
78
NO
IF MCRERR = 3 YES
SET UP PARAMETERS FOR LINE TROUBLE MESSAGE
78
NO
IF MCRERR = 4 YES
SET UP PARAMETERS FOR DATA TOO LONG MESSAGE
78
NO
IF MCRERR = 5 YES
DISPLAY ILLOGICAL REQUEST MESSAGE
12
NO
11

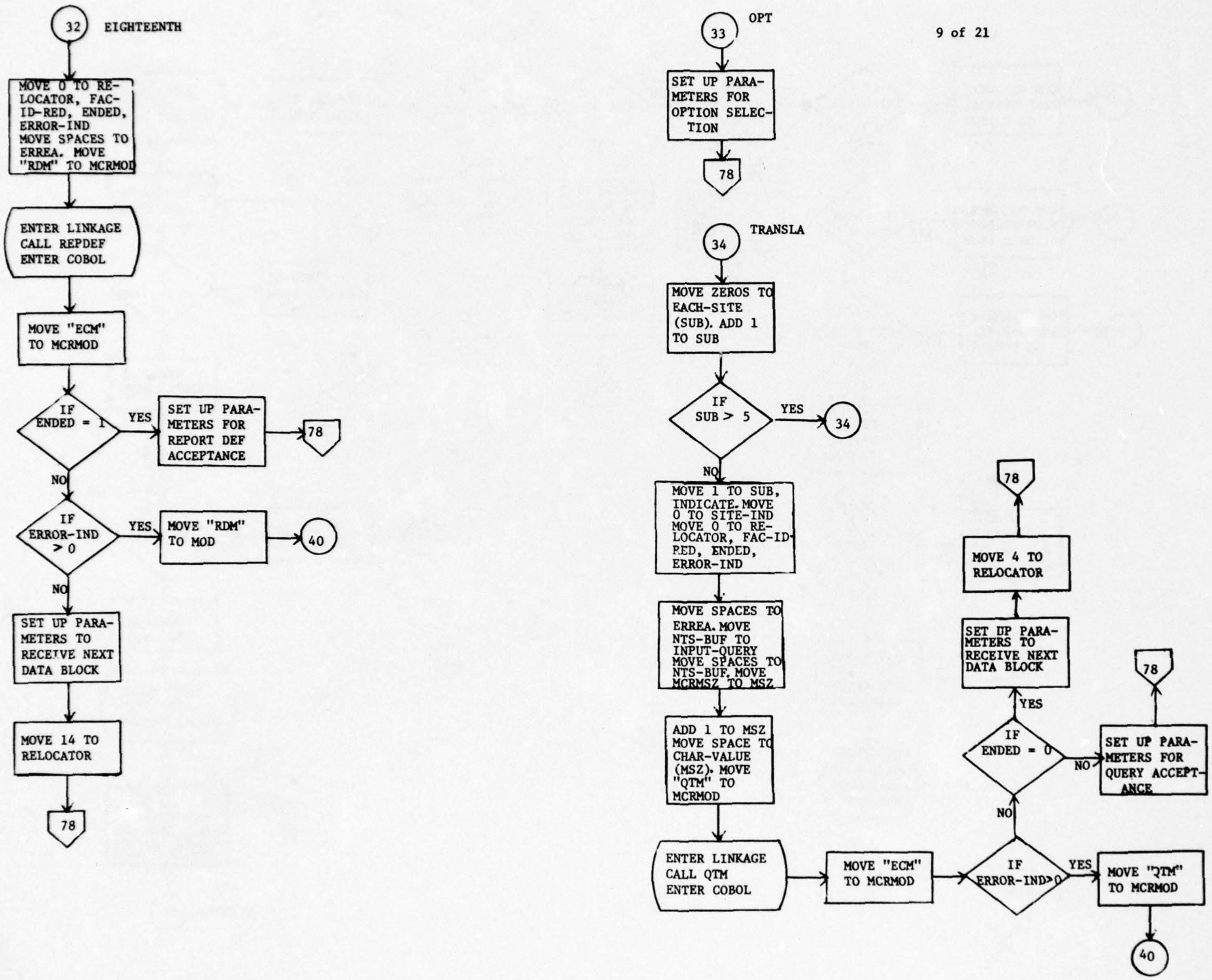

17
FINAL-BLK
*6
IF STORE-ERR = SPACES
YES
MOVE SPACES TO STORE-ERR, ERREA. MOVE STORE-ERR TO LINE 1. MOVE "RCM" TO MOD
40
NO
IF RSVP-OPEN = 1
YES
54
NO
IF MCRERR > 0
YES
MOVE 13 TO RELOCATOR
16
NO
IF FAC-ID = "000001"
YES
34
NO
GO TO DEPENDING ON RELOCATOR
18
IMMEDIATE-RETURN
GO TO DEPENDING ON MCRRET
SET UP PARAMETERS FOR NO INDICATOR MESSAGE
78
*6
IF ECM-RCM = RCM AND CONSTR-OPEN =1
YES
E
NO
19
GET-FIRST
*7
SET UP PARAMETERS FOR FACILITY ID MESSAGE
MOVE 1 TO RELOCATOR
78
*7
MOVE ECM TO ECM-RCM
20
FIRST/SECOND/THIRD/FOURTH
SET UP PARAMETERS FOR PASSWORD BLACK OUT
MOVE 3 TO RELOCATOR
78

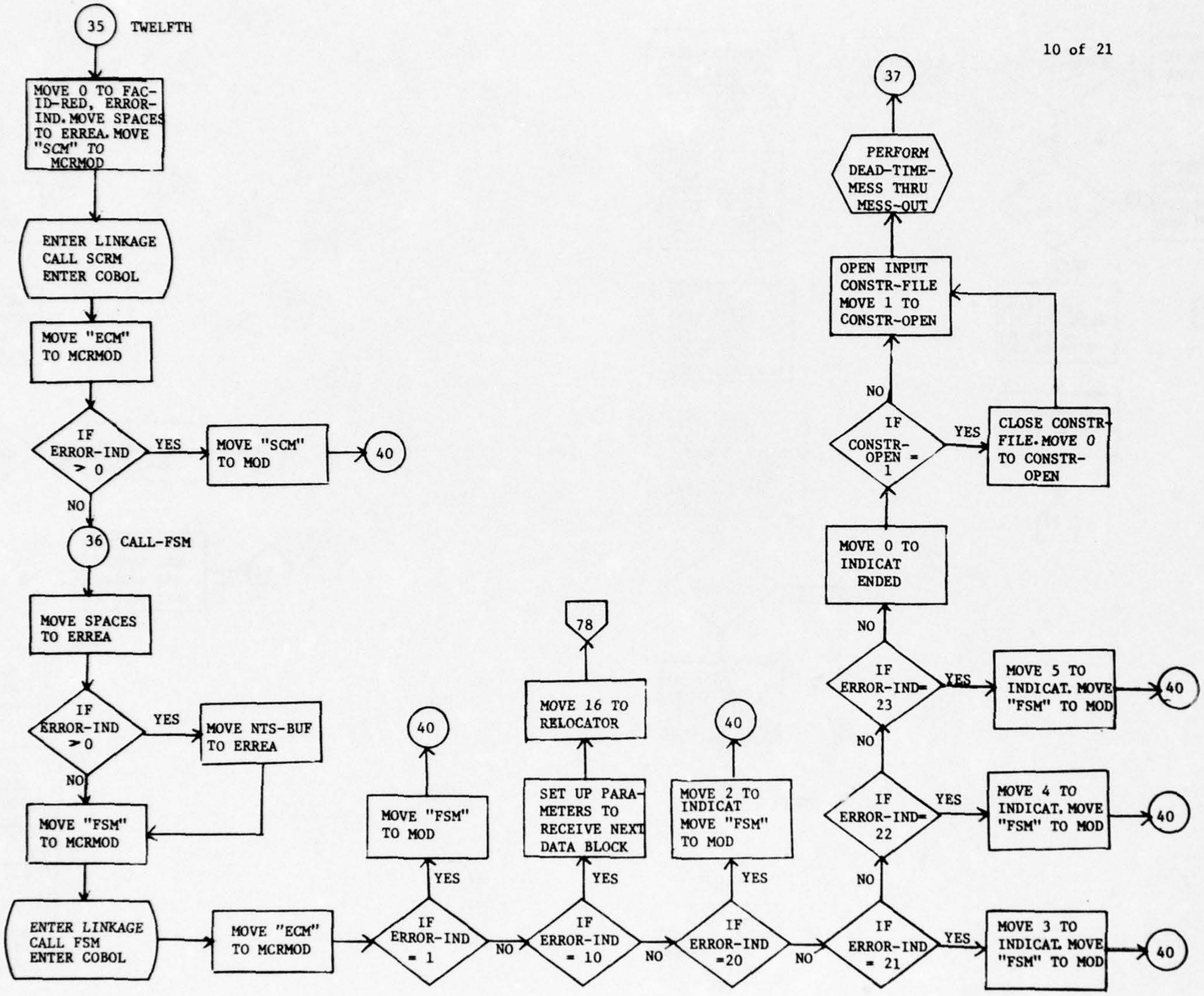


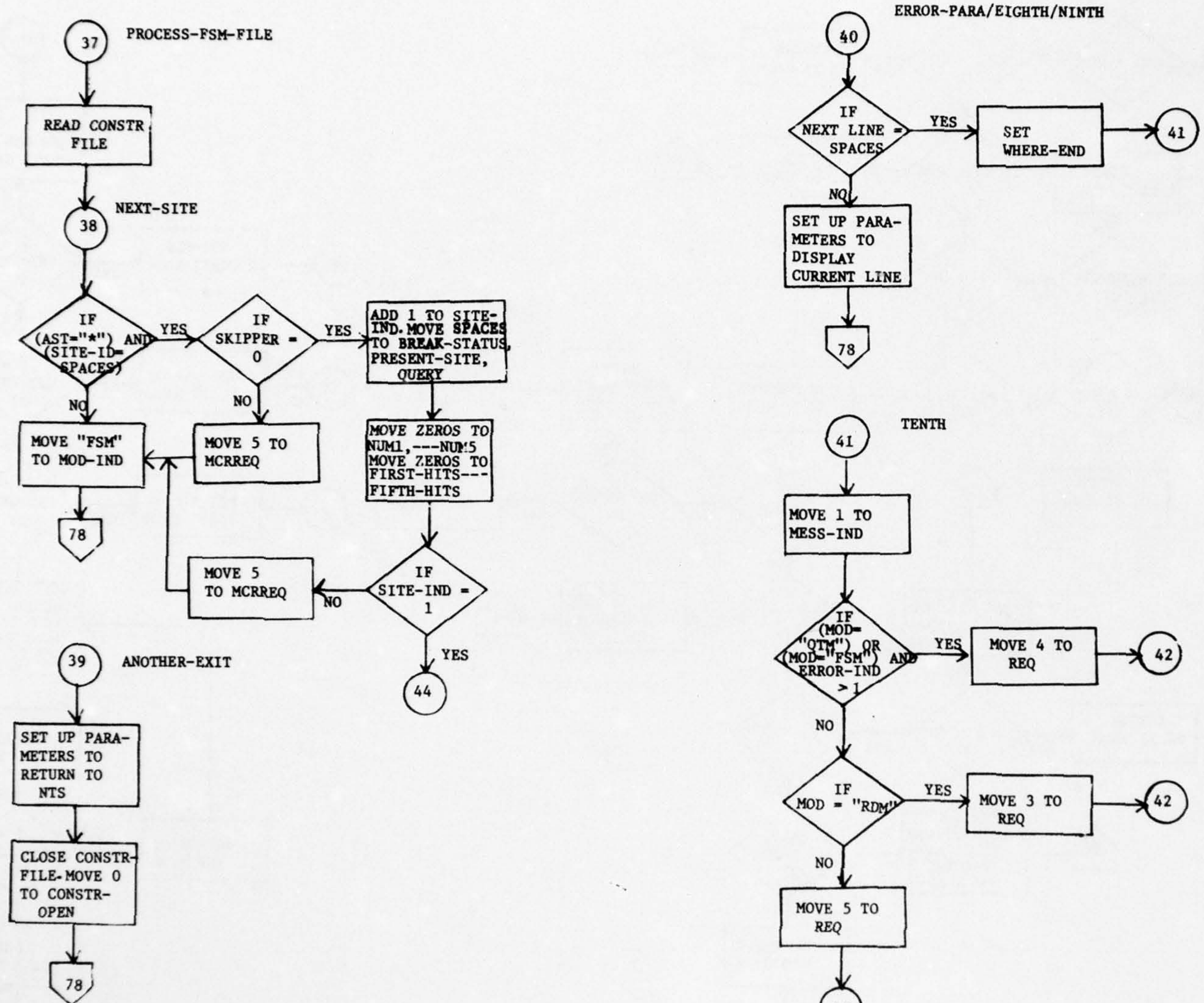
22
TRY-AGAIN
SET UP PARA- METERS FOR GET HELP MESSAGE
MOVE 0 TO ATTEMPTS, VALIDATOR
78
23
THIRTEENTH
SET UP PARA- METERS TO DISCONNECT LINE
78
21
FAC-CK
MOVE 0 TO RELO- CATOR. MOVE NTS-BUF TO FAC- ID. MOVE SPACES TO NTS-BUF
IF FAC-ID IS NUMERIC
IF FAC-ID = "000001"
YES
SET UP PARA- METERS TO IN- DICATE TRANS- FER COMPLETE
78
NO
MOVE 1 TO RSVP-OPEN. OPEN OUTPUT RSVP-FILE MOVE ZEROS TO SITE-STATUS- TABLE
MOVE "DPSPAC" TO FILE-IDENT OPEN INPUT EDIT-FILE
GO TO DEPEND- ING ON FAC-ID- RED
NO
PERFORM FAC-VAL THRU END- FAC-VAL
IF VALID
YES
SET UP PARA- METERS FOR USER ID MESSAGE
MOVE 2 TO RELOCATOR
78
NO
ADD 1 TO ATTEMPTS
IF ATTEMPTS < 3
NO
MOVE 0 TO VALIDATOR
22
YES
SET UP PARA- METERS FOR INVALID FACILITY ID
MOVE 1 TO RELOCATOR MOVE 0 TO VALIDATOR
78

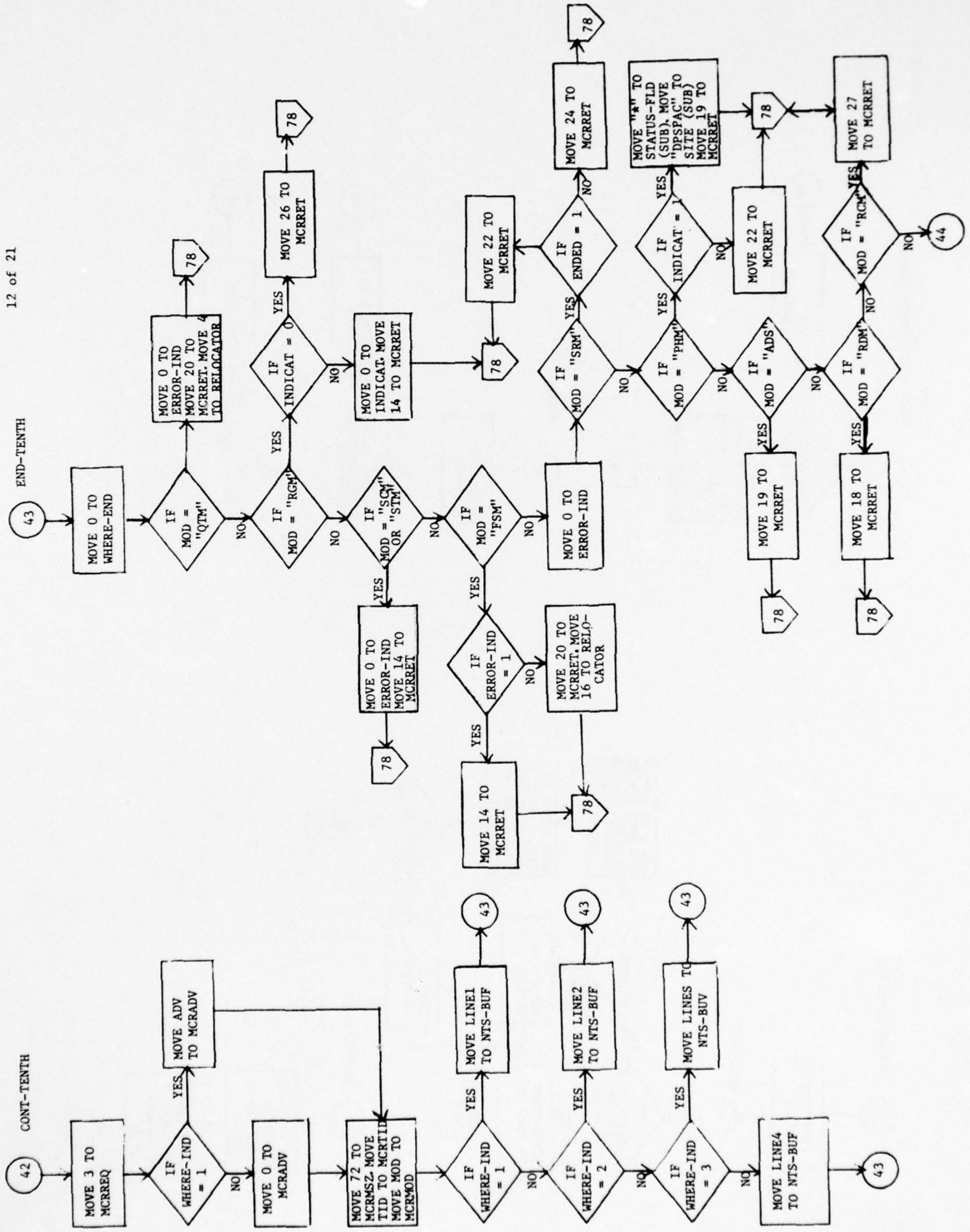

24
USER-CK
MOVE 0 TO RELOCATOR MOVE NTS-BUF TO USER-ID MOVE SPACES TO NTS-BUF
PERFORM USER-VAL THRU END-USER-VAL
IF VALID
YES
SET UP PARA-METERS FOR PASSWORD ENTRY MESSAGE
MOVE 0 TO ATTEMPTS
78
NO
ADD 1 TO ATTEMPTS
IF ATTEMPTS < 3
YES
SET UP PARA-METERS FOR INVALID USER ID MESSAGE
MOVE 2 TO RELOCATOR MOVE 0 TO VALIDATOR
78
NO
MOVE 0 TO VALIDATOR
22
25
PASSWRD-CK
MOVE 0 TO RELOCATOR MOVE NTS-BUF TO PASSWRD
PERFORM PASSWRD-VAL THRU END-PASSWRD-VAL
IF VALID
YES
SET UP PARA-METERS FOR OPTION SELEC-TION
MOVE 8 TO RELOCATOR MOVE 0 TO ATTEMPTS
78
NO
ADD 1 TO ATTEMPTS
IF ATTEMPTS < 3
YES
SET UP PARA-METERS FOR ILLEGAL PASSWORD
MOVE 0 TO VALIDATOR
78
NO
MOVE 0 TO VALIDATOR MOVE 3 TO RELOCATOR
22

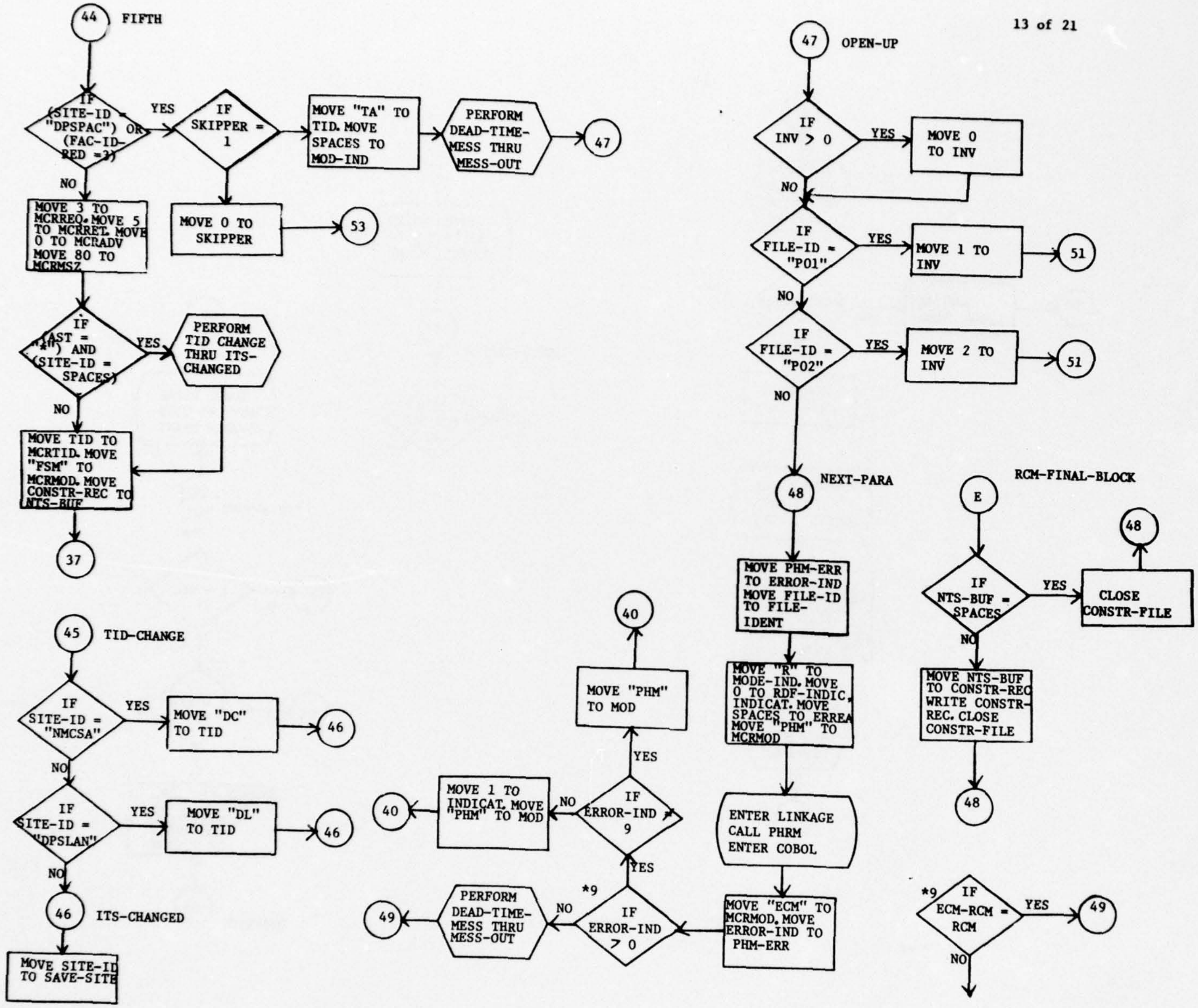

FIND-SITE
26
READ CONSTR-FILE
IF AST = "*"
YES
NO
*8
IF SITE-ID = SAVE-SITE
YES
NO
27
FOUND
EXIT
26
FIFTEENTH
28
CLOSE EDIT-FILE, OPEN INPUT EDIT-FILE
61
SIXTEENTH
29
IF CONSTR-OPEN = 1
YES
CLOSE CONSTR-FILE. MOVE 0 TO CONSTR-OPEN
NO
OPEN INPUT CONSTR-FILE MOVE 1 TO CONSTR-OPEN
PERFORM FIND-SITE THRU FOUND
MOVE 1 TO SKIPPER
44
*8
IF SITE-ID= SAVE-SITE OR ECM-RCM= RCM
YES
NO
OPTION
30
MOVE 0 TO RELOCATOR, IN-LOGIN MOVE NTS-BUF TO COMMANDER MOVE SPACES TO NTS-BUF
IF COMMANDER = "END"
YES
67
NO
IF COMMANDER = "REPDEF"
YES
SET UP PARAMETERS FOR REPORT DEF INPUT
MOVE 14 TO RELOCATOR
78
NO
SET UP PARAMETERS FOR QUERY INPUT
MOVE 4 TO RELOCATOR
78
CALL-REPDEF
31
MOVE NTS-BUF TO RDF-IN MOVE SPACES TO NTS-BUF MOVE MCRMSZ TO MSZ
32

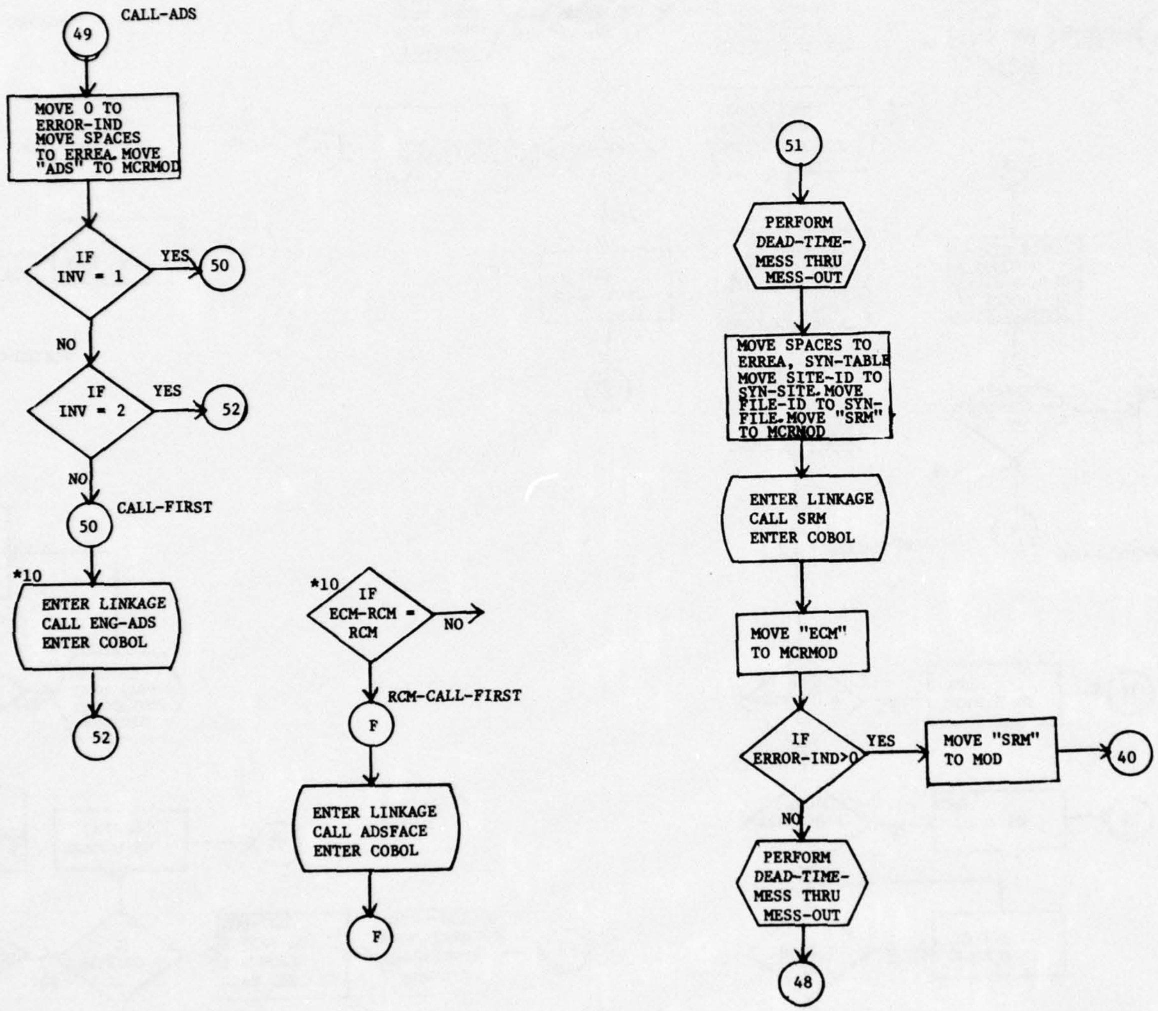

32
EIGHTEENTH
MOVE 0 TO RE-LOCATOR, FAC-ID-RED, ENDED, ERROR-IND MOVE SPACES TO ERREA. MOVE "RDM" TO MCRMOD
ENTER LINKAGE CALL REPDEF ENTER COBOL
MOVE "ECM" TO MCRMOD
IF ENDED = 1
YES
SET UP PARA-METERS FOR REPORT DEF ACCEPTANCE
78
NO
IF ERROR-IND > 0
YES
MOVE "RDM" TO MOD
40
NO
SET UP PARA-METERS TO RECEIVE NEXT DATA BLOCK
MOVE 14 TO RELOCATOR
78
33
OPT
SET UP PARA-METERS FOR OPTION SELEC-TION
78
34
TRANSLA
MOVE ZEROS TO EACH-SITE (SUB). ADD 1 TO SUB
IF SUB > 5
YES
34
NO
MOVE 1 TO SUB, INDICATE. MOVE 0 TO SITE-IND MOVE 0 TO RE-LOCATOR, FAC-ID-RED, ENDED, ERROR-IND
MOVE SPACES TO ERREA. MOVE NTS-BUF TO INPUT-QUERY MOVE SPACES TO NTS-BUF. MOVE MCRMSZ TO MSZ
ADD 1 TO MSZ MOVE SPACE TO CHAR-VALUE (MSZ). MOVE "QTM" TO MCRMOD
ENTER LINKAGE CALL QTM ENTER COBOL
MOVE "ECM" TO MCRMOD
IF ERROR-IND>0
YES
MOVE "QTM" TO MCRMOD
40
NO
IF ENDED = 0
YES
SET UP PARA-METERS TO RECEIVE NEXT DATA BLOCK
MOVE 4 TO RELOCATOR
78
NO
SET UP PARA-METERS FOR QUERY ACCEPT-ANCE
78

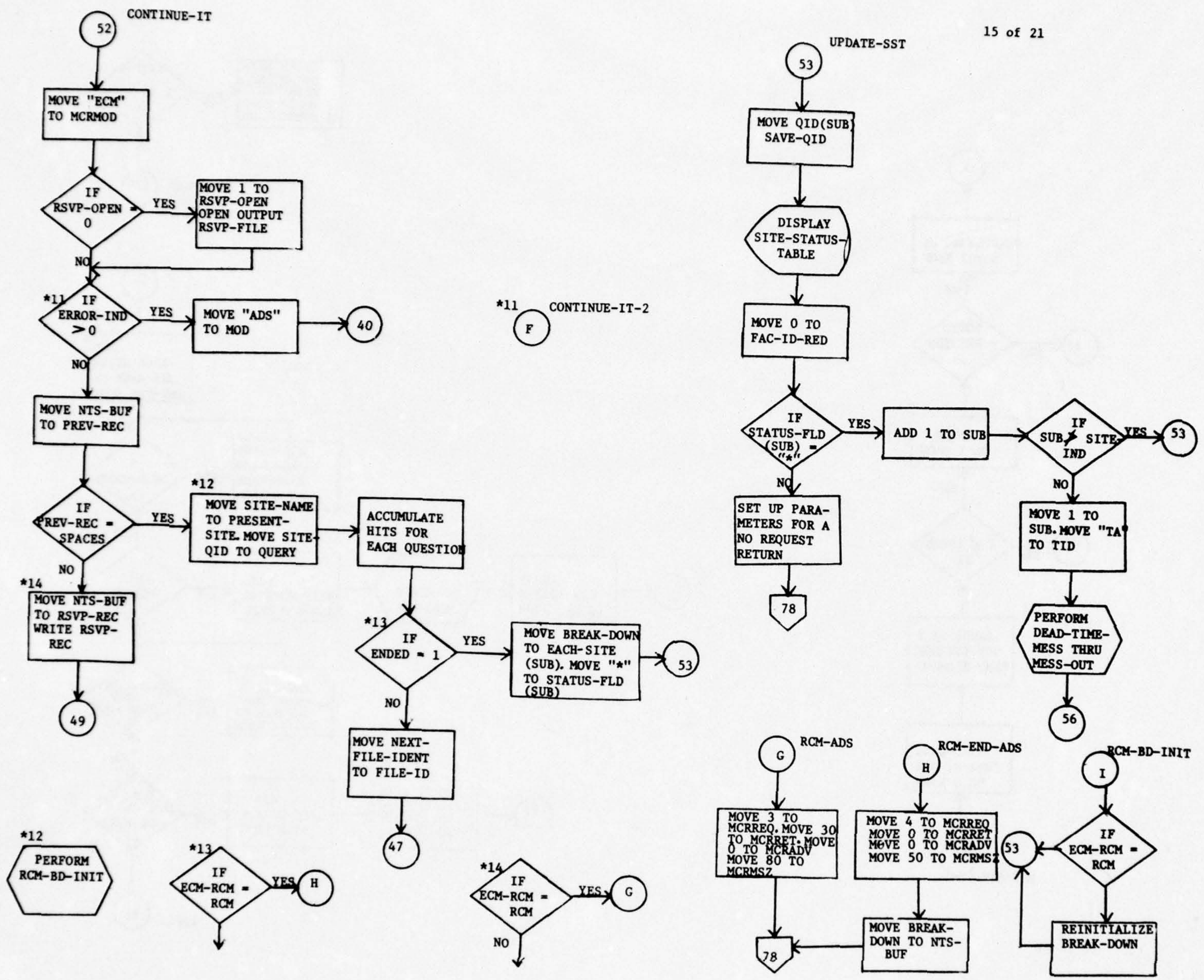

35
TWELFTH
MOVE 0 TO FAC-ID-RED, ERROR-IND. MOVE SPACES TO ERREA. MOVE "SCM" TO MCRMOD
ENTER LINKAGE CALL SCRM ENTER COBOL
MOVE "ECM" TO MCRMOD
IF ERROR-IND > 0
YES
MOVE "SCM" TO MOD
40
NO
36
CALL-FSM
MOVE SPACES TO ERREA
IF ERROR-IND > 0
YES
MOVE NTS-BUF TO ERREA
NO
MOVE "FSM" TO MCRMOD
ENTER LINKAGE CALL FSM ENTER COBOL
MOVE "ECM" TO MCRMOD
IF ERROR-IND = 1
YES
MOVE "FSM" TO MOD
40
NO
IF ERROR-IND = 10
YES
SET UP PARAMETERS TO RECEIVE NEXT DATA BLOCK
MOVE 16 TO RELOCATOR
78
NO
IF ERROR-IND =20
YES
MOVE 2 TO INDICAT MOVE "FSM" TO MOD
40
NO
IF ERROR-IND = 21
YES
MOVE 3 TO INDICAT. MOVE "FSM" TO MOD
40
NO
IF ERROR-IND= 22
YES
MOVE 4 TO INDICAT. MOVE "FSM" TO MOD
40
NO
IF ERROR-IND= 23
YES
MOVE 5 TO INDICAT. MOVE "FSM" TO MOD
40
NO
MOVE 0 TO INDICAT ENDED
IF CONSTR-OPEN = 1
YES
CLOSE CONSTR-FILE. MOVE 0 TO CONSTR-OPEN
NO
OPEN INPUT CONSTR-FILE MOVE 1 TO CONSTR-OPEN
PERFORM DEAD-TIME-MESS THRU MESS-OUT
37

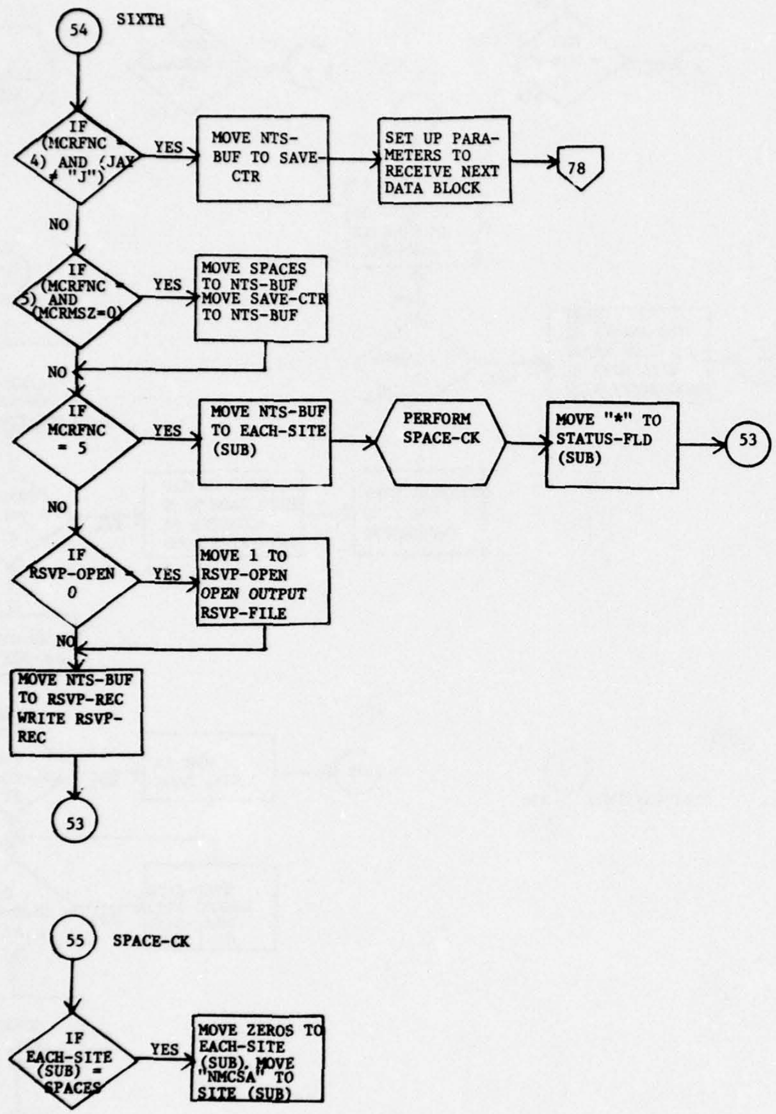

37
PROCESS-FSM-FILE
READ CONSTR FILE
NEXT-SITE
38
IF (AST="*") AND (SITE-ID= SPACES)
YES
NO
MOVE "FSM" TO MOD-IND
78
IF SKIPPER = 0
YES
NO
MOVE 5 TO MCRREQ
ADD 1 TO SITE-IND. MOVE SPACES TO BREAK-STATUS, PRESENT-SITE, QUERY
MOVE ZEROS TO NUM1,---NUM5 MOVE ZEROS TO FIRST-HITS---FIFTH-HITS
IF SITE-IND = 1
NO
MOVE 5 TO MCRREQ
YES
44
39
ANOTHER-EXIT
SET UP PARAMETERS TO RETURN TO NTS
CLOSE CONSTR-FILE-MOVE 0 TO CONSTR-OPEN
78
ERROR-PARA/EIGHTH/NINTH
40
IF NEXT LINE = SPACES
YES
SET WHERE-END
41
NO
SET UP PARAMETERS TO DISPLAY CURRENT LINE
78
TENTH
41
MOVE 1 TO MESS-IND
IF (MOD="QTM") OR (MOD="FSM") AND ERROR-IND > 1
YES
MOVE 4 TO REQ
42
NO
IF MOD = "RDM"
YES
MOVE 3 TO REQ
42
NO
MOVE 5 TO REQ
42

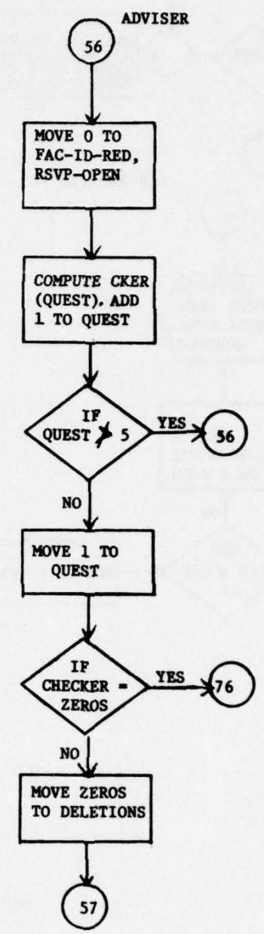

END-TENTH
43
MOVE 0 TO WHERE-END
IF MOD = "QTM"
MOVE 0 TO ERROR-IND MOVE 20 TO MCRRET. MOVE 4 TO RELOCATOR
78
IF MOD = "RGM"
IF INDICAT = 0
MOVE 26 TO MCRRET
MOVE 0 TO INDICAT. MOVE 14 TO MCRRET
IF MOD = "SCM" OR "STM"
MOVE 0 TO ERROR-IND MOVE 14 TO MCRRET
IF MOD = "FSM"
IF ERROR-IND = 1
MOVE 14 TO MCRRET
MOVE 20 TO MCRRET. MOVE 16 TO RELO-CATOR
MOVE 0 TO ERROR-IND
MOVE 22 TO MCRRET
IF MOD = "SRM"
IF ENDED = 1
MOVE 24 TO MCRRET
IF MOD = "PHM"
IF INDICAT = 1
MOVE "A" TO STATUS-FLD (SUB). MOVE "DPSPAC" TO SITE (SUB) MOVE 19 TO MCRRET
MOVE 22 TO MCRRET
IF MOD = "ADS"
MOVE 19 TO MCRRET
IF MOD = "RDM"
MOVE 18 TO MCRRET
IF MOD = "RCM"
MOVE 27 TO MCRRET
44
YES
NO
CONT-TENTH
42
MOVE 3 TO MCRREQ
IF WHERE-IND = 1
MOVE ADV TO MCRADV
MOVE 0 TO MCRADV
MOVE 72 TO MCRMSZ. MOVE TID TO MCRTID MOVE MOD TO MCRMOD
IF WHERE-IND = 1
MOVE LINE1 TO NTS-BUF
IF WHERE-IND = 2
MOVE LINE2 TO NTS-BUF
IF WHERE-IND = 3
MOVE LINES TO NTS-BUV
MOVE LINE4 TO NTS-BUF

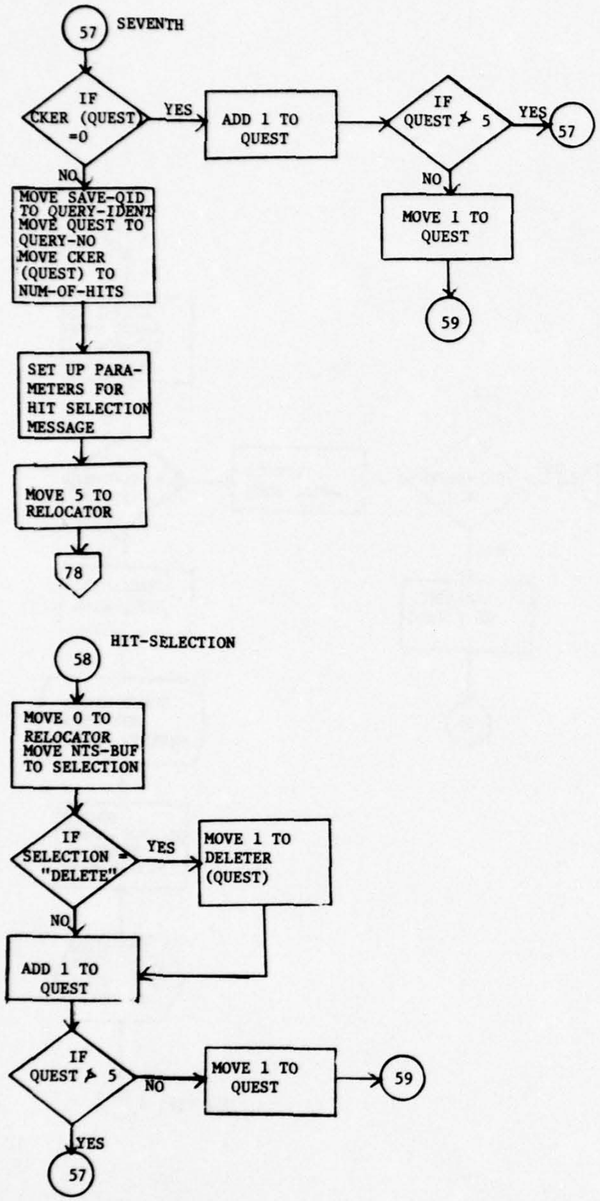

44 FIFTH
IF (SITE-ID = "DPSPAC") OR (FAC-ID-RED =3)
YES
NO
IF SKIPPER = 1
MOVE "TA" TO TID. MOVE SPACES TO MOD-IND
PERFORM DEAD-TIME-MESS THRU MESS-OUT
47
MOVE 0 TO SKIPPER
53
MOVE 3 TO MCRREQ. MOVE 5 TO MCRRET. MOVE 0 TO MCRADV MOVE 80 TO MCRMSZ
IF (AST = "*") AND (SITE-ID = SPACES)
PERFORM TID CHANGE THRU ITS-CHANGED
MOVE TID TO MCRTID. MOVE "FSM" TO MCRMOD. MOVE CONSTR-REC TO NTS-BUF
37
45 TID-CHANGE
IF SITE-ID = "NMCSA"
MOVE "DC" TO TID
46
IF SITE-ID = "DPSLAN"
MOVE "DL" TO TID
46 ITS-CHANGED
MOVE SITE-ID TO SAVE-SITE
47 OPEN-UP
IF INV > 0
MOVE 0 TO INV
IF FILE-ID = "P01"
MOVE 1 TO INV
51
IF FILE-ID = "P02"
MOVE 2 TO INV
48 NEXT-PARA
MOVE PHM-ERR TO ERROR-IND MOVE FILE-ID TO FILE-IDENT
MOVE "R" TO MODE-IND. MOVE 0 TO RDF-INDIC, INDICAT. MOVE SPACES TO ERREA MOVE "PHM" TO MCRMOD
ENTER LINKAGE CALL PHRM ENTER COBOL
MOVE "ECM" TO MCRMOD. MOVE ERROR-IND TO PHM-ERR
*9
IF ERROR-IND > 0
PERFORM DEAD-TIME-MESS THRU MESS-OUT
49
IF ERROR-IND ≠ 9
MOVE "PHM" TO MOD
40
MOVE 1 TO INDICAT. MOVE "PHM" TO MOD
E RCM-FINAL-BLOCK
IF NTS-BUF = SPACES
CLOSE CONSTR-FILE
MOVE NTS-BUF TO CONSTR-REC WRITE CONSTR-REC. CLOSE CONSTR-FILE
IF ECM-RCM = RCM

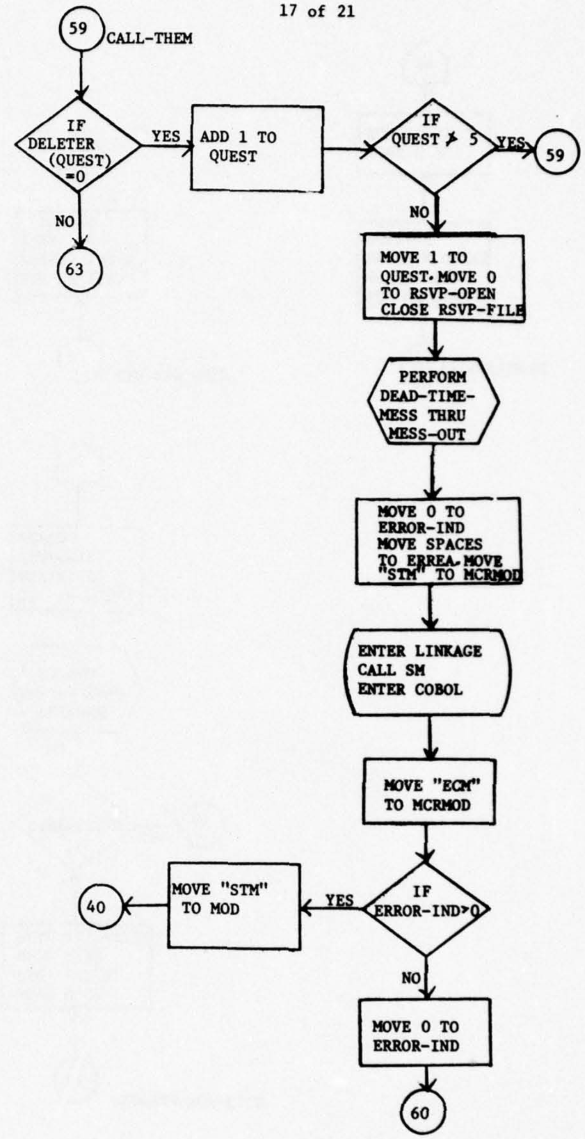

CALL-ADS
49
MOVE 0 TO ERROR-IND MOVE SPACES TO ERREA. MOVE "ADS" TO MCRMOD
IF INV = 1
YES
50
NO
IF INV = 2
YES
52
NO
CALL-FIRST
50
*10
ENTER LINKAGE CALL ENG-ADS ENTER COBOL
52
*10
IF ECM-RCM = RCM
NO
RCM-CALL-FIRST
F
ENTER LINKAGE CALL ADSFACE ENTER COBOL
F
51
PERFORM DEAD-TIME-MESS THRU MESS-OUT
MOVE SPACES TO ERREA, SYN-TABLE MOVE SITE-ID TO SYN-SITE. MOVE FILE-ID TO SYN-FILE. MOVE "SRM" TO MCRMOD
ENTER LINKAGE CALL SRM ENTER COBOL
MOVE "ECM" TO MCRMOD
IF ERROR-IND>0
YES
MOVE "SRM" TO MOD
40
NO
PERFORM DEAD-TIME-MESS THRU MESS-OUT
48

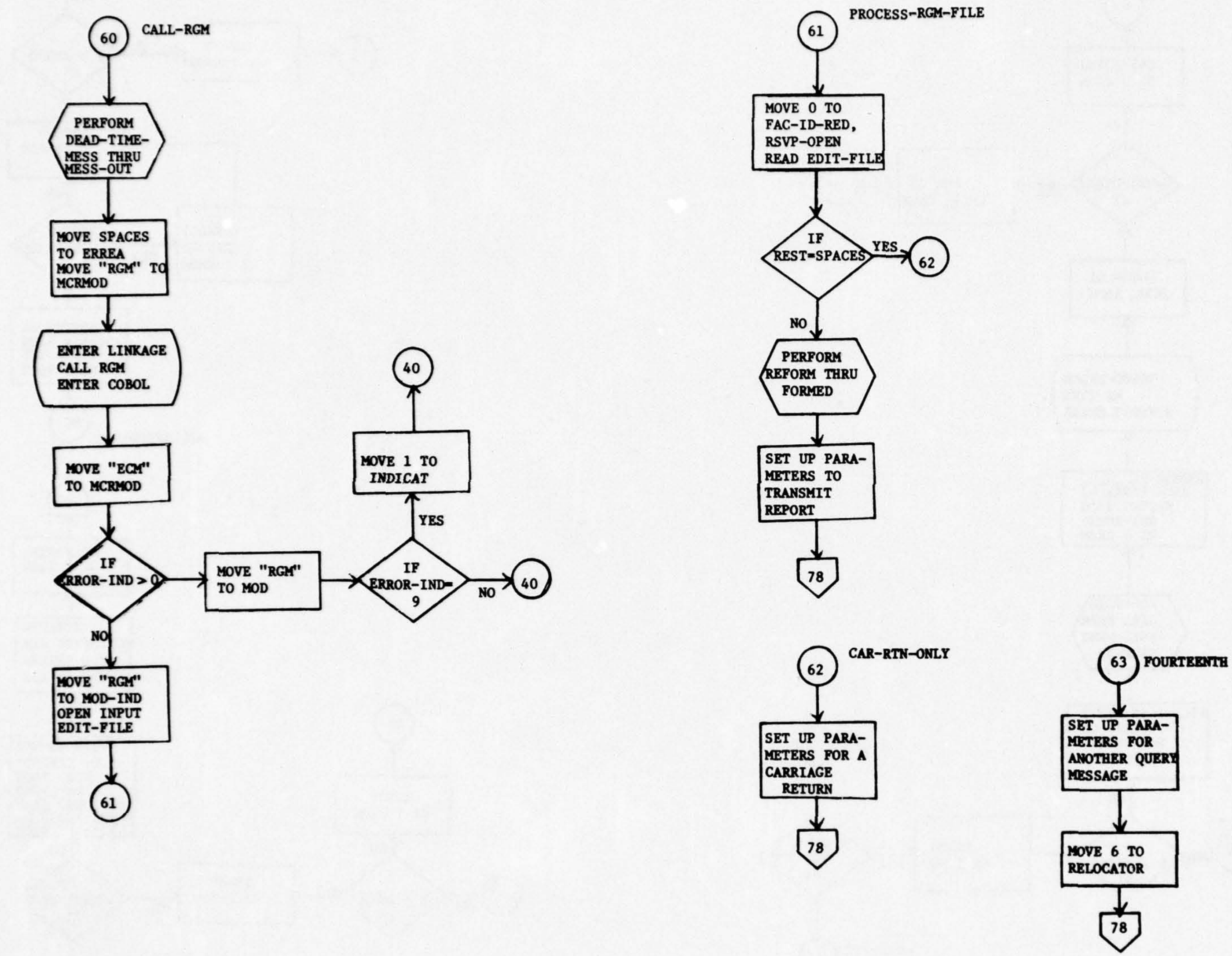

CONTINUE-IT
52
MOVE "ECM" TO MCRMOD
IF RSVP-OPEN = 0
YES
MOVE 1 TO RSVP-OPEN OPEN OUTPUT RSVP-FILE
NO
*11
IF ERROR-IND > 0
YES
MOVE "ADS" TO MOD
40
NO
MOVE NTS-BUF TO PREV-REC
IF PREV-REC = SPACES
YES
*12
MOVE SITE-NAME TO PRESENT-SITE. MOVE SITE-QID TO QUERY
ACCUMULATE HITS FOR EACH QUESTION
*13
IF ENDED = 1
YES
MOVE BREAK-DOWN TO EACH-SITE (SUB). MOVE "*" TO STATUS-FLD (SUB)
53
NO
MOVE NEXT-FILE-IDENT TO FILE-ID
47
NO
*14
MOVE NTS-BUF TO RSVP-REC WRITE RSVP-REC
49
*11
F
CONTINUE-IT-2
*12
PERFORM RCM-BD-INIT
*13
IF ECM-RCM = RCM
YES
H
*14
IF ECM-RCM = RCM
YES
G
NO
UPDATE-SST
53
MOVE QID(SUB) SAVE-QID
DISPLAY SITE-STATUS-TABLE
MOVE 0 TO FAC-ID-RED
IF STATUS-FLD (SUB) = "*"
YES
ADD 1 TO SUB
IF SUB ≠ SITE-IND
YES
53
NO
MOVE 1 TO SUB. MOVE "TA" TO TID
PERFORM DEAD-TIME-MESS THRU MESS-OUT
56
NO
SET UP PARAMETERS FOR A NO REQUEST RETURN
78
G
RCM-ADS
MOVE 3 TO MCRREQ. MOVE 30 TO MCRRET. MOVE 0 TO MCRADV MOVE 80 TO MCRMSZ
78
H
RCM-END-ADS
MOVE 4 TO MCRREQ MOVE 0 TO MCRRET MOVE 0 TO MCRADV MOVE 50 TO MCRMSZ
MOVE BREAK-DOWN TO NTS-BUF
I
RCM-BD-INIT
IF ECM-RCM = RCM
53
REINITIALIZE BREAK-DOWN

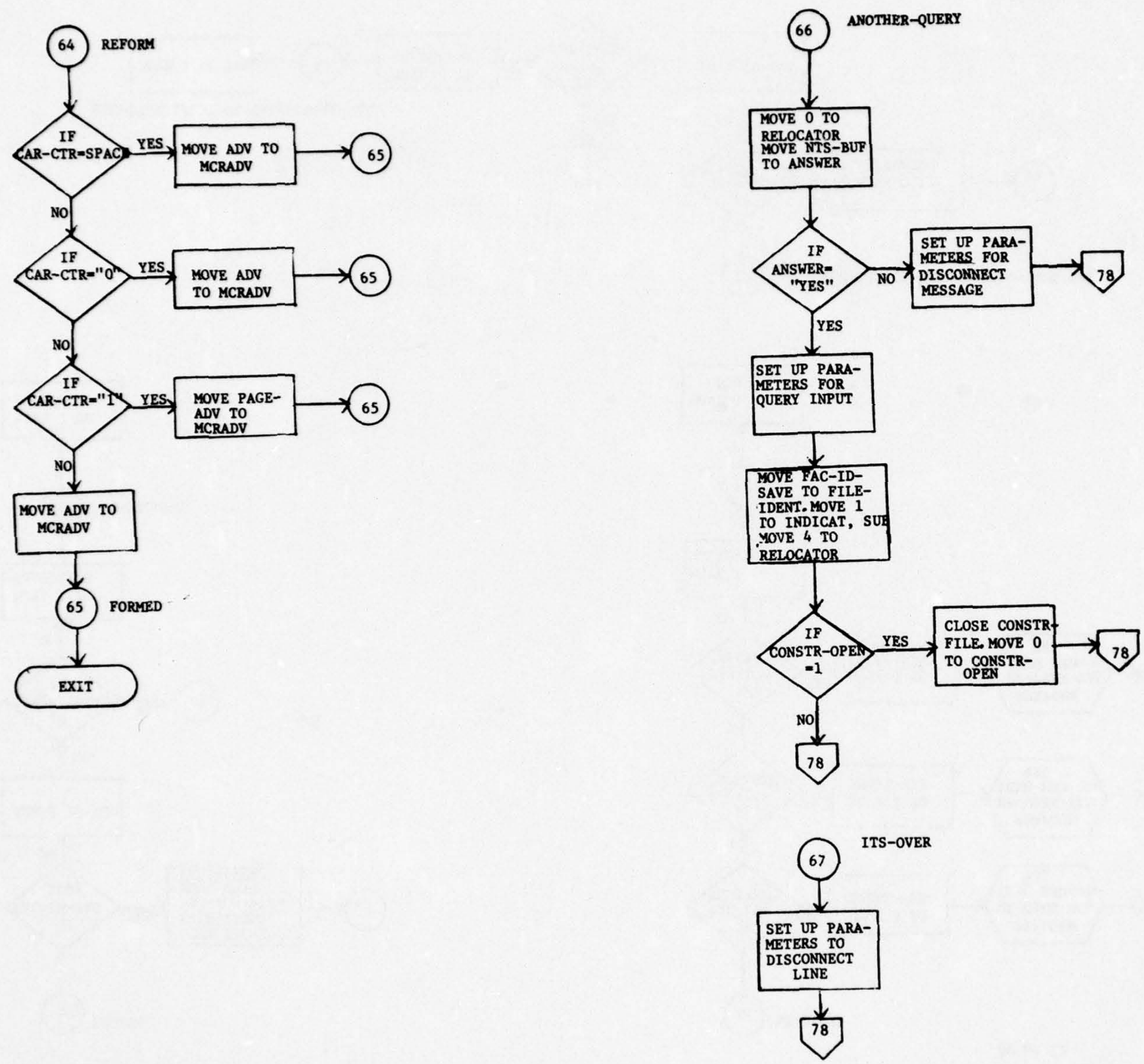

SIXTH
54
IF (MCRFNC = 4) AND (JAY ≠ "J")
YES
MOVE NTS-BUF TO SAVE-CTR
SET UP PARAMETERS TO RECEIVE NEXT DATA BLOCK
78
NO
IF (MCRFNC = 5) AND (MCRMSZ=0)
YES
MOVE SPACES TO NTS-BUF MOVE SAVE-CTR TO NTS-BUF
NO
IF MCRFNC = 5
YES
MOVE NTS-BUF TO EACH-SITE (SUB)
PERFORM SPACE-CK
MOVE "*" TO STATUS-FLD (SUB)
53
NO
IF RSVP-OPEN = 0
YES
MOVE 1 TO RSVP-OPEN OPEN OUTPUT RSVP-FILE
NO
MOVE NTS-BUF TO RSVP-REC WRITE RSVP-REC
53
55
SPACE-CK
IF EACH-SITE (SUB) = SPACES
YES
MOVE ZEROS TO EACH-SITE (SUB), MOVE "NMCSA" TO SITE (SUB)

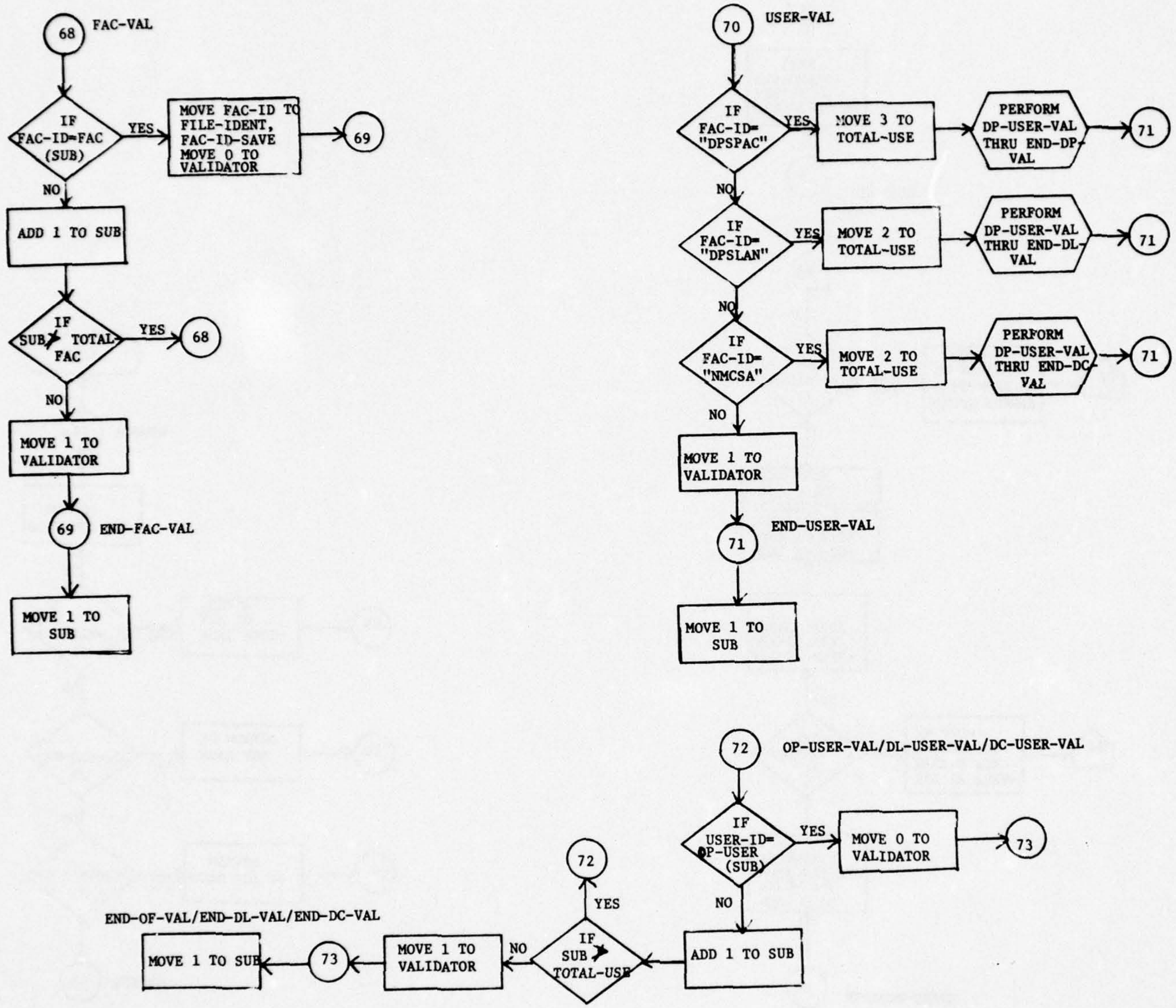

ADVISER
56
MOVE 0 TO FAC-ID-RED, RSVP-OPEN
COMPUTE CKER (QUEST), ADD 1 TO QUEST
IF QUEST ≯ 5
YES
56
NO
MOVE 1 TO QUEST
IF CHECKER = ZEROS
YES
76
NO
MOVE ZEROS TO DELETIONS
57


57
SEVENTH
IF CKER (QUEST) =0
YES
ADD 1 TO QUEST
IF QUEST ≯ 5
YES
57
NO
MOVE 1 TO QUEST
59
NO
MOVE SAVE-QID TO QUERY-IDENT MOVE QUEST TO QUERY-NO MOVE CKER (QUEST) TO NUM-OF-HITS
SET UP PARA-METERS FOR HIT SELECTION MESSAGE
MOVE 5 TO RELOCATOR
78
58
HIT-SELECTION
MOVE 0 TO RELOCATOR MOVE NTS-BUF TO SELECTION
IF SELECTION = "DELETE"
YES
MOVE 1 TO DELETER (QUEST)
NO
ADD 1 TO QUEST
IF QUEST ≯ 5
NO
MOVE 1 TO QUEST
59
YES
57

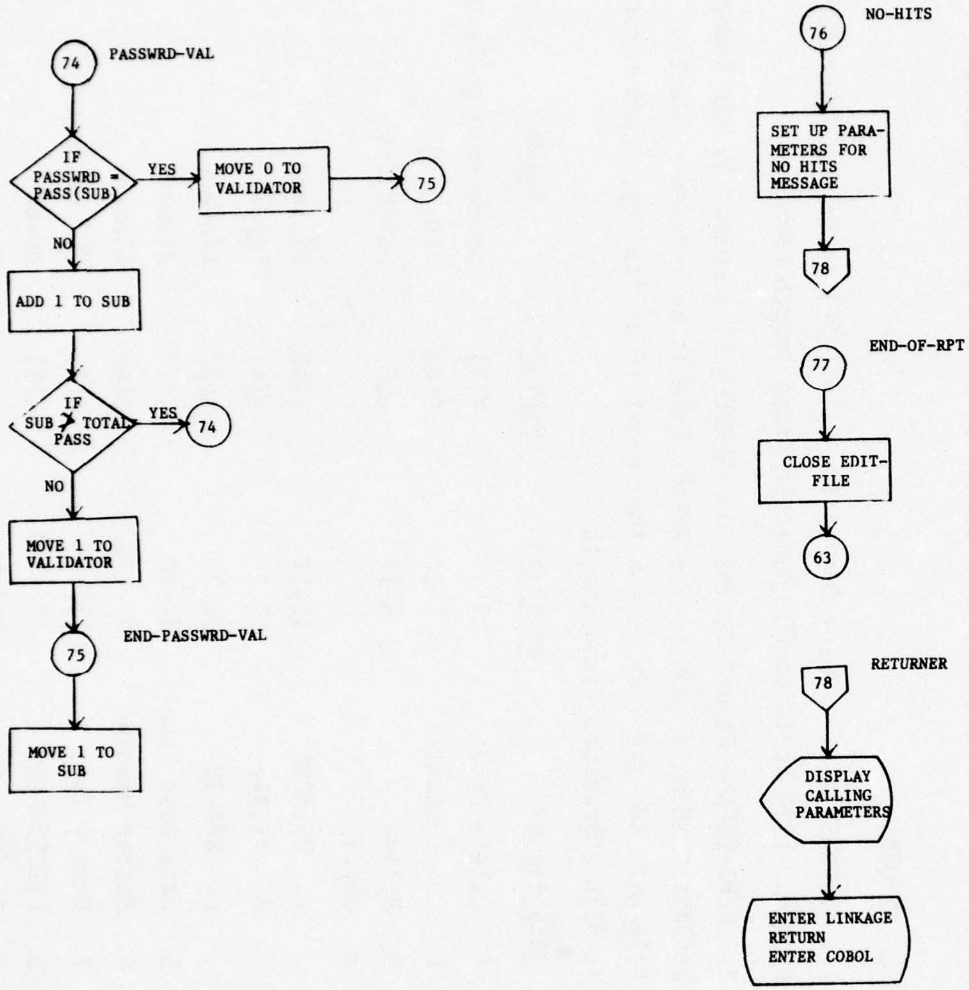

59
CALL-THEM
IF DELETER (QUEST) =0
YES
ADD 1 TO QUEST
IF QUEST ≯ 5
YES
59
NO
63
NO
MOVE 1 TO QUEST. MOVE 0 TO RSVP-OPEN CLOSE RSVP-FILE
PERFORM DEAD-TIME-MESS THRU MESS-OUT
MOVE 0 TO ERROR-IND MOVE SPACES TO ERREA. MOVE "STM" TO MCRMOD
ENTER LINKAGE CALL SM ENTER COBOL
MOVE "ECM" TO MCRMOD
IF ERROR-IND>0
YES
MOVE "STM" TO MOD
40
NO
MOVE 0 TO ERROR-IND
60


60
CALL-RGM
PERFORM DEAD-TIME-MESS THRU MESS-OUT
MOVE SPACES TO ERREA MOVE "RGM" TO MCRMOD
ENTER LINKAGE CALL RGM ENTER COBOL
MOVE "ECM" TO MCRMOD
IF ERROR-IND > 0
NO
MOVE "RGM" TO MOD-IND OPEN INPUT EDIT-FILE
61
MOVE "RGM" TO MOD
IF ERROR-IND= 9
YES
MOVE 1 TO INDICAT
40
NO
40
61
PROCESS-RGM-FILE
MOVE 0 TO FAC-ID-RED, RSVP-OPEN READ EDIT-FILE
IF REST=SPACES
YES
62
NO
PERFORM REFORM THRU FORMED
SET UP PARAMETERS TO TRANSMIT REPORT
78
62
CAR-RTN-ONLY
SET UP PARAMETERS FOR A CARRIAGE RETURN
78
63
FOURTEENTH
SET UP PARAMETERS FOR ANOTHER QUERY MESSAGE
MOVE 6 TO RELOCATOR
78


64
REFORM
IF CAR-CTR=SPACE
YES
MOVE ADV TO MCRADV
65
NO
IF CAR-CTR="0"
YES
MOVE ADV TO MCRADV
65
NO
IF CAR-CTR="1"
YES
MOVE PAGE-ADV TO MCRADV
65
NO
MOVE ADV TO MCRADV
65
FORMED
EXIT
66
ANOTHER-QUERY
MOVE 0 TO RELOCATOR MOVE NTS-BUF TO ANSWER
IF ANSWER="YES"
NO
SET UP PARAMETERS FOR DISCONNECT MESSAGE
78
YES
SET UP PARAMETERS FOR QUERY INPUT
MOVE FAC-ID-SAVE TO FILE-IDENT. MOVE 1 TO INDICAT, SUB MOVE 4 TO RELOCATOR
IF CONSTR-OPEN =1
YES
CLOSE CONSTR-FILE. MOVE 0 TO CONSTR-OPEN
78
NO
78
67
ITS-OVER
SET UP PARAMETERS TO DISCONNECT LINE
78



68
FAC-VAL
IF FAC-ID=FAC (SUB)
YES
MOVE FAC-ID TO FILE-IDENT, FAC-ID-SAVE MOVE 0 TO VALIDATOR
69
NO
ADD 1 TO SUB
IF SUB > TOTAL FAC
YES
68
NO
MOVE 1 TO VALIDATOR
69
END-FAC-VAL
MOVE 1 TO SUB
70
USER-VAL
IF FAC-ID= "DPSPAC"
YES
MOVE 3 TO TOTAL-USE
PERFORM DP-USER-VAL THRU END-DP-VAL
71
NO
IF FAC-ID= "DPSLAN"
YES
MOVE 2 TO TOTAL-USE
PERFORM DP-USER-VAL THRU END-DL-VAL
71
NO
IF FAC-ID= "NMCSA"
YES
MOVE 2 TO TOTAL-USE
PERFORM DP-USER-VAL THRU END-DC-VAL
71
NO
MOVE 1 TO VALIDATOR
71
END-USER-VAL
MOVE 1 TO SUB
72
OP-USER-VAL/DL-USER-VAL/DC-USER-VAL
IF USER-ID= OP-USER (SUB)
YES
MOVE 0 TO VALIDATOR
73
NO
ADD 1 TO SUB
IF SUB > TOTAL-USE
YES
72
NO
MOVE 1 TO VALIDATOR
73
END-OF-VAL/END-DL-VAL/END-DC-VAL
MOVE 1 TO SUB



74
PASSWRD-VAL
IF PASSWRD = PASS(SUB)
YES
MOVE 0 TO VALIDATOR
75
NO
ADD 1 TO SUB
IF SUB > TOTAL PASS
YES
74
NO
MOVE 1 TO VALIDATOR
75
END-PASSWRD-VAL
MOVE 1 TO SUB
76
NO-HITS
SET UP PARAMETERS FOR NO HITS MESSAGE
78
77
END-OF-RPT
CLOSE EDIT-FILE
63
78
RETURNER
DISPLAY CALLING PARAMETERS
ENTER LINKAGE
RETURN
ENTER COBOL

A.3 TITLE: Screening Module (SCRM)

A.3.1 PURPOSE

The Screening Module (SCRM) makes modifications, needed by other modules, to the Query Constraint File (QCF) which was output by the Query Translator Module (QTM). These modifications may involve reformatting, adding, and/or deleting records. Changes are made in SCRM because at this point the QCF has not been expanded by the File Selection Module (FSM), and therefore duplication of effort will not result.

A.3.2 INPUTS

1. IN-CONSTRAINT - Query Constraint File (QCF) as it was generated by QTM. File contains 80-character fixed length records.

2. REPT-DEFN - Report Definition (REPDEF) File output by the Report Definition Module (RDM). The REPDEF File is an indexed sequential file with the facility ID and the report ID as its key. Each record is 3110 characters fixed length.

Data Field	Position	Picture	Value
1. Delete Code	1	X(1)	Space or High-Value
2. Char. Count	2-5	9(4)	110-3110
3. Filler	6-11	X(6)	Asterisks
4. RDF-ID			
a. DB-NAME	12-21	X(10)	Alphan.
b. Filler	22-27	X(6)	"Report"
c. RPT-ID	28-30	X(3)	Alphan.
5. Data Base Type	31-40	X(10)	Alphan.
6. Master Heading	41-110	X(60)	Alphan.
7. Report Type	111	X(1)	"C" or Space
8. LIRCS/PAGE	112-114	9(3)	00-60
9. Page Width	115-117	9(3)	60
10. RDF Block	occurs 100 times		
a. LIRC		X(4)	X(4) Numeric
b. RDF INFO-1		X(17)	X(17) Alphan.
c. Comp. Code		X(1)	X(1) "A"...."H"
d. RDF INFO-2		X(8)	X(8) Alphan.

A.3.3 OUTPUT

OUT-CONSTRAINT - QCF as revised by SCRM. Records are still 80 characters fixed length. Records may have been changed, added, and/or deleted as specified in Section A.3.6 (PROCESSING DESCRIPTION). Record type is indicated in parentheses following record name.

A. Site/File ID Record (*)

	Data Field	Position	Picture	Value
1.	Asterisk	1	X(1)	"*"
2.	Site ID	2-7	X(6)	Alphan.
3.	File ID	8-10	X(3)	Alphan.
4.	File Type	11	X(1)	"R" or "S"

B. Merge Record (A)

	Data Field	Position	Picture	Value
1.	Query ID	1-3	X(3)	Alphan.
2.	Question No.	4-6	X(3)	Numeric
3.	Filler	7-8	X(2)	Spaces
4.	Type Code	9	X(1)	"A"
5.	Set No.	10-11	X(2)	00 or 01
6.	Filler	12-80	X(69)	Spaces

C. Report Selection Record (B)

	Data Field	Position	Picture	Value
1.	Query ID	1-3	X(3)	Alphan.
2.	Question No.	4-6	X(3)	Numeric
3.	Filler	7-8	X(2)	Spaces
4.	Type Code	9	X(1)	"B"
5.	Set No.	10-11	X(2)	Numeric
6.	Filler	12	X(1)	Space
7.	Literal	13-18	X(6)	"Report"
8.	Filler	19	X(1)	Space
9.	Facility ID	20-25	X(6)	Alphan.

10.	Filler	26	X(1)	Space
11.	Report ID	27-29	X(3)	Alphan.
12.	Filler	30-80	X(51)	Spaces

D. Computation Record (C)

	Data Field	Position	Picture	Value
1.	Query ID	1-3	X(3)	Alphan.
2.	Question No.	4-6	X(3)	Numeric
3.	Filler	7	X(1)	Space
4.	Control In	8	X(1)	"K","N", or "P"
5.	Type Code	9	X(1)	"C"
6.	Set No.	10-11	X(2)	Numeric
7.	Comp. Code	12	X(1)	"A"..."H"
8.	LIRC-1	13-16	X(4)	Numeric
9.	Filler	17	X(1)	Space
10.	Acronym-1	18-27	X(10)	Alphan.
11.	Filler	28	X(1)	Space
12.	Form(Type)-1	29	X(1)	I,R,N,A,L, or Space
13.	Decimal-1	30	X(1)	Numeric or Space
14.	Filler	31-32	X(2)	Spaces
* 15.	LIRC-2	33-36	X(4)	Numeric
16.	Filler	37	X(1)	Space
* 17.	Acronym-2	38-47	X(10)	Alphan.
18.	Filler	48	X(1)	Space
* 19.	Form(Type)-2	49	X(1)	I,R,N,A,L, or Space
* 20.	Decimal-2	50	X(1)	Numeric or Space
21.	Filler	51-80	X(30)	Spaces

* Only for certain Comp. Codes

E. Sort Record (D)

	Data Field	Position	Picture	Value
1.	Query ID	1-3	X(3)	Alphan.
2.	Question No.	4-6	X(3)	Numeric

3.	Filler	7-8	X(2)	Spaces
4.	Type Code	9	X(1)	"D"
5.	Set No.	10-11	X(2)	Numeric
6.	High/Low	12	X(1)	H,L, or Space
7.	LIRC	13-16	X(4)	Numeric
8.	Prefix/Partial	17-26	X(10)	[1](NUM) or *NUM (NUM)
9.	Char. Count	27-30	9(4)	Numeric
10.	Filler	31-80	X(50)	Spaces

[1]NUM = 000 - 999

F. Variable Heading Record (H)

	Data Field	Position	Picture	Value
1.	Query ID	1-3	X(3)	Alphan.
2.	Question No.	4-6	X(3)	Numeric
3.	Filler	7-8	X(2)	Spaces
4.	Type Code	9	X(1)	"H"
5.	Set No.	10-11	X(2)	Numeric
6.	Filler	12	X(1)	Space
7.	Variable Heading	13-72	X(60)	Alphan.
8.	Filler	73-80	X(8)	Spaces

G. Selection Record (J)

	Data Field	Position	Picture	Value
1.	Query ID	1-3	X(3)	Alphan.
2.	Question No.	4-6	X(3)	Numeric
3.	Filler	7-8	X(2)	Spaces
4.	Type Code	9	X(1)	"J"
5.	Set No.	10-11	X(2)	Numeric
6.	Filler	12	X(1)	Space
7.	LIRC	13-16	X(4)	Numeric
8.	Filler	17-26	X(10)	Spaces
9.	Parallel Ind.	27	X(1)	"P" or Space
10.	Filler	28-80	X(53)	Spaces

H. Constraint Record (L-Z)

	Data Field	Position	Picture	Value
1.	Query ID	1-3	X(3)	Alphan.
2.	Question No.	4-6	X(3)	Numeric
3.	OP Code	7-8	X(2)	"BE","BH","BL", "BN","EQ","GE", "GR","LE","LS", or "NE"
4.	Type Code	9	X(1)	"L"..."Z"
5.	Set No.	10-11	X(2)	Numeric
* 6.	Mod(L/R)	12	X(1)	Alphan.
* 7.	Link	13	X(1)	Alphan.
* 8.	Role	14	X(1)	Alphan.
9.	Search Type	15	X(1)	"W","X","P", or "S"
10.	LIRC	16-19	X(4)	Numeric
* 11.	No. of Links	20-21	9(2)	Numeric
* 12.	No. of Sets	22-24	9(3)	Numeric
13.	Inversion Indic.	25-26	X(2)	Numeric or Spaces
14.	Parallel LIRC Indic.	27	X(1)	"P" or Space
15.	Efficiency Code	28	X(1)	"X" or Space
16.	Search Value(s)			
	a. Non-Ranging Search	29-78	X(50)	Alphan.
	b. Ranging Search			
	1) Value-1	29-53	X(25)	Alphan.
	2) Value-2	54-78	X(25)	Alphan.
17.	Filler	79-80	X(2)	Spaces

* Currently not used in NAVLIS System, treat as filler.

I. Parallel LIRC Record (9)

	Data Field	Position	Picture	Value
1.	Query ID	1-3	X(3)	Alphan.
2.	Question No.	4-6	X(3)	Numeric
3.	Filler	7-8	X(2)	Spaces
4.	Type Code	9	X(1)	"9"
5.	Set No.	10-11	X(2)	Numeric
6.	Filler	12-15	X(4)	Spaces
7.	LIRC	16-19	X(4)	Numeric
8.	Filler	20	X(1)	Space
9.	Parallel LIRC #1	21-24	X(4)	Numeric or Spaces
10.	Filler	25	X(1)	Space
11.	Parallel LIRC #2	26-29	X(4)	Numeric or Spaces
12.	Filler	30	X(1)	Space
13.	Parallel LIRC #3	31-34	X(4)	Numeric or Spaces
14.	Filler	35	X(1)	Space
15.	Parallel LIRC #4	36-39	X(4)	Numeric or Spaces
16.	Filler	40	X(1)	Space
17.	Parallel LIRC #5	41-44	X(4)	Numeric or Spaces
18.	Filler	45	X(1)	Space
19.	Parallel LIRC #6	46-49	X(4)	Numeric or Spaces
20.	Filler	50	X(1)	Space
21.	Parallel LIRC #7	51-54	X(4)	Numeric or Spaces
22.	Filler	55	X(1)	Space
23.	Parallel LIRC #8	56-59	X(4)	Numeric or Spaces
24.	Filler	60	X(1)	Space
25.	Parallel LIRC #9	61-64	X(4)	Numeric or Spaces
26.	Filler	65	X(1)	Space
27.	Parallel LIRC #10	66-69	X(11)	Spaces

A.3.4 RECORD FORMATS - See Figures A1-A5 and A8.

A.3.5 STORAGE REQUIREMENTS:

Program (less library modules) occupies 27,000 bytes on UNIVAC Series 70/45.

Fold back at dotted line.

Section & Page 101.
Subsection
Date Documented
Change Notice #

CARD OR RECORD LAYOUT – DOUBLE

SYSTEM ID: ☐ New ☐ Revision

RECORD TITLE:
FILE ID: REPORT DEFINITION FILE
RECORD LENGTH: 147 MINIMUM ☑ FIXED ☐ 3110 MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR 1 RECORD/BLOCK

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Positions	Field	Note
1	DELETE CODE	(1)
2–5	CHAR COUNT	(2)
6–11	FILLER	(3)
12–29	RDF-ID	
12–21	DB-NAME	(4)
22–27	FILLER	(5)
28–29	RPT ID	(6)
30–39	DATA BASE TYPE	(7)
40–100	MASTER HEADING	(8)

RECORD TITLE:
FILE ID: REPORT DEFINITION FILE (CONTINUED)
RECORD LENGTH: MINIMUM ☐ FIXED ☐ MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☐
BLOCKING FACTOR

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Positions	Field	Note
11	RPT TYPE	(9)
12–13	LINES/PAGE	(10)
14–17	WIDTH	(11)
18–47	RDF BLOCK OCCURS 100 TIMES	
18–21	LIRC	(12)
22–38	RDF INFO 1	(13)
39	COMP	(14)
40–47	RDF INFO 2	(15)

(1) SPACE OR HIGH VALUE
(2) "110" --- "3110"
(3) ASTERISKS
(4) ALPHANUMERIC
(5) "REPORT "
(6) ALPHANUMERIC
(7) ALPHANUMERIC
(8) ALPHANUMERIC
(9) "C" OR SPACE
(10) "00" --- "60"
(11) "60"
(12) NUMERIC
(13) ALPHANUMERIC
(14) "A" --- "H"
(15) ALPHANUMERIC

Figure A-8

A.3.6 PROCESSING DESCRIPTION (Refer to SCRM Flowchart, page 55)

A.3.6.1 GENERAL DESCRIPTION

SCRM creates the new QCF by reading and storing the old QCF records and processing the different record types for each question. Whenever all the constraints for a question have been stored, SCRM will do some final checking on parallelism and on adding J-type Logistic Information Requirements Codes (LIRC). Then all the stored records for the question will be written to the new QCF. The above processing continues for all questions until an end-of-file is encountered on QCF.

A.3.6.2 LZ ROUTINE

For all L through Z constraints, SCRM first determines whether the search type is prefix (X). If it is prefix and the operation code (op code) is NE, then all spaces of the search value will be replaced by asterisks (*). If, however, the op code is not NE, then SCRM changes the search type to whole (w) and replaces the spaces in the search value(s) with low-values or high-values, whichever is appropriate. For example, a greater than (GR) search on value ABC (spaces) will become a GR search on value ABC (low-values). If, however, the op code is EQ, SCRM will change it to BE and move the value of the first search value into the second search value field. Then, all the spaces in the first value will be changed to low-values and those of the second value to high-values. After the prefix checking for a record is done, the efficiency indicator is checked.

If the efficiency indicator is on (X), then the op code is checked. If it is a range operator (BE,BH,BL, or BN), then an efficiency search cannot be performed on the constraint so the efficiency indicator is reset to space. For all other op codes, the original op code will be replaced by its complement, (e.g. NE becomes EQ, GR becomes LE, etc.). Then the L through Z record is stored in the LZ table.

A.3.6.3 CJB ROUTINE (also includes ADH processing)

For all C or J records, SCRM stores the record in the CJ table. If it is a J record, then the J flag is set to 1. For a B record, SCRM sets the B flag to 1, stores the B record, and creates the REPDEF key using the facility ID and the report ID. SCRM stores the remaining records (A,D, or H) in the ADH table.

A.3.6.4 PARALLEL PROCESSING

When a type 9 record is encountered while reading the QCF, SCRM determines whether any parallelism has already been found. If not, SCRM checks the newest 9 record for parallelism.

Parallelism exits when two or more L-Z LIRCs are parallel to each other. (The two LIRCs may be identical.) Each L-Z LIRC which is parallel to other LIRCs causes a type 9 record to be created on the QCF in QTM. The 9 record contains, first, the LIRC which was on the L-Z constraint (the QCF LIRC) and then all the LIRCs parallel to it in numerical order. Parallelism is such that if LIRC A is parallel to LIRC B, then LIRC B is also parallel to LIRC A. Furthermore, if LIRC C is parallel to LIRC A, it is also parallel to LIRC B. Therefore, the 9 records for LIRCs which are parallel to one another will contain the same LIRCs. The only difference is which LIRC is the QCF LIRC. Also, either the QCF LIRC or the first parallel LIRC will have the minimum value (numerical) of all LIRCs on the record. Once the minimum LIRC has been determined, parallelism can be found by checking the minimum LIRC of each 9 record against the minimum LIRCs of the 9 records already read and stored. If a match is found, the parallel indicator is set to the index value of the stored 9 record which is parallel. Otherwise, the new 9 record and its minimum LIRC are stored with the previous ones and will be used in checking for parallelism if more 9 records are encountered.

A.3.6.5 END OF QUESTION PROCESSING

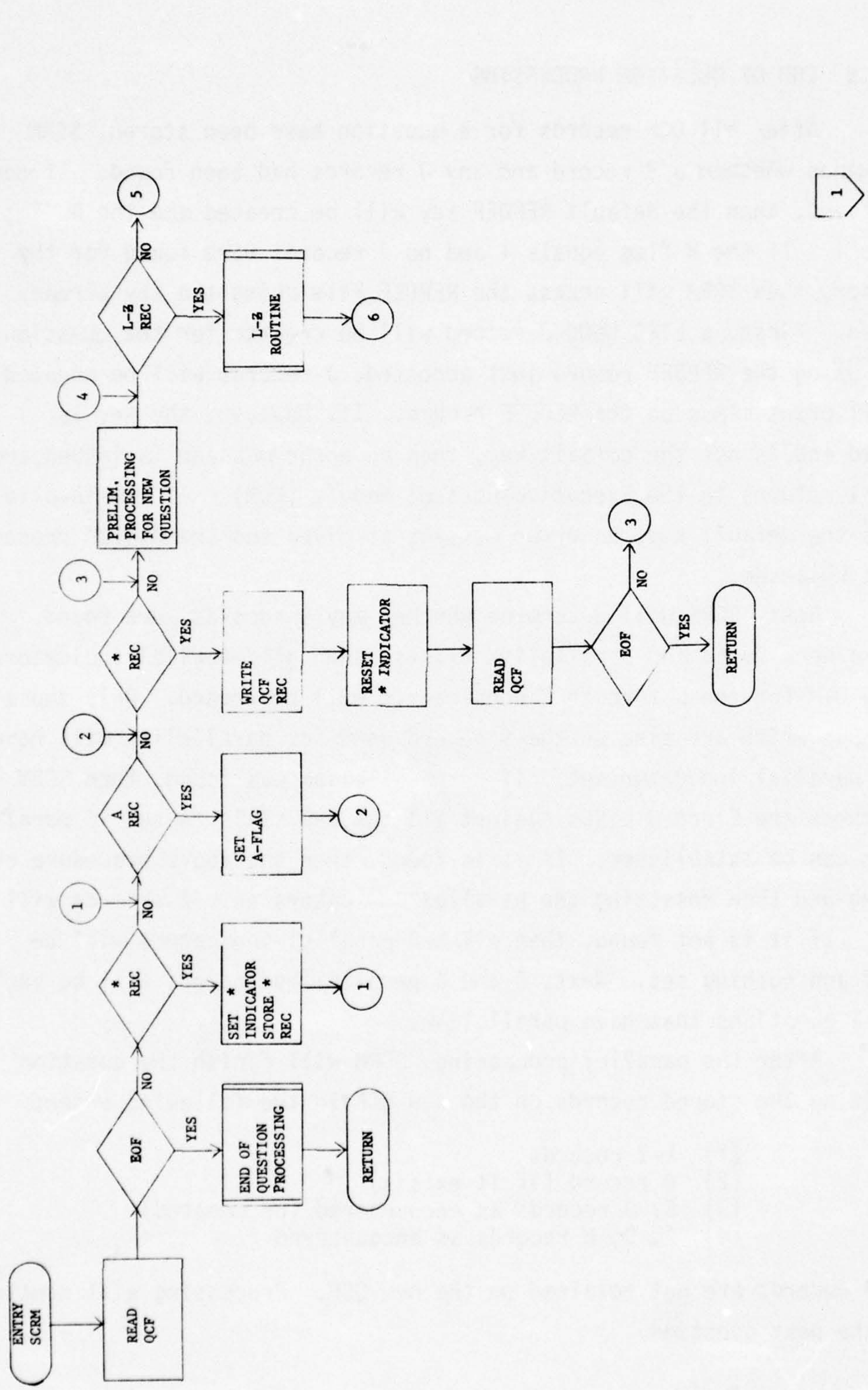
After all QCF records for a question have been stored, SCRM determines whether a B record and any J records had been found. If none were found, then the default REPDEF key will be created and the B flag set to 1. If the B flag equals 1 and no J records were found for the question, then SCRM will access the REPDEF File using the key already created. First, a LIRC 0000 J record will be created for the question. Then, using the REPDEF record just accessed, J records will be created for all print LIRCs on the REPDEF record. If, however, the key is invalid and is not the default key, then an error message is issued and control returns to the Executive Control Module (ECM). If the invalid key is the default key, no error message is given and the J LIRC processing is bypassed.

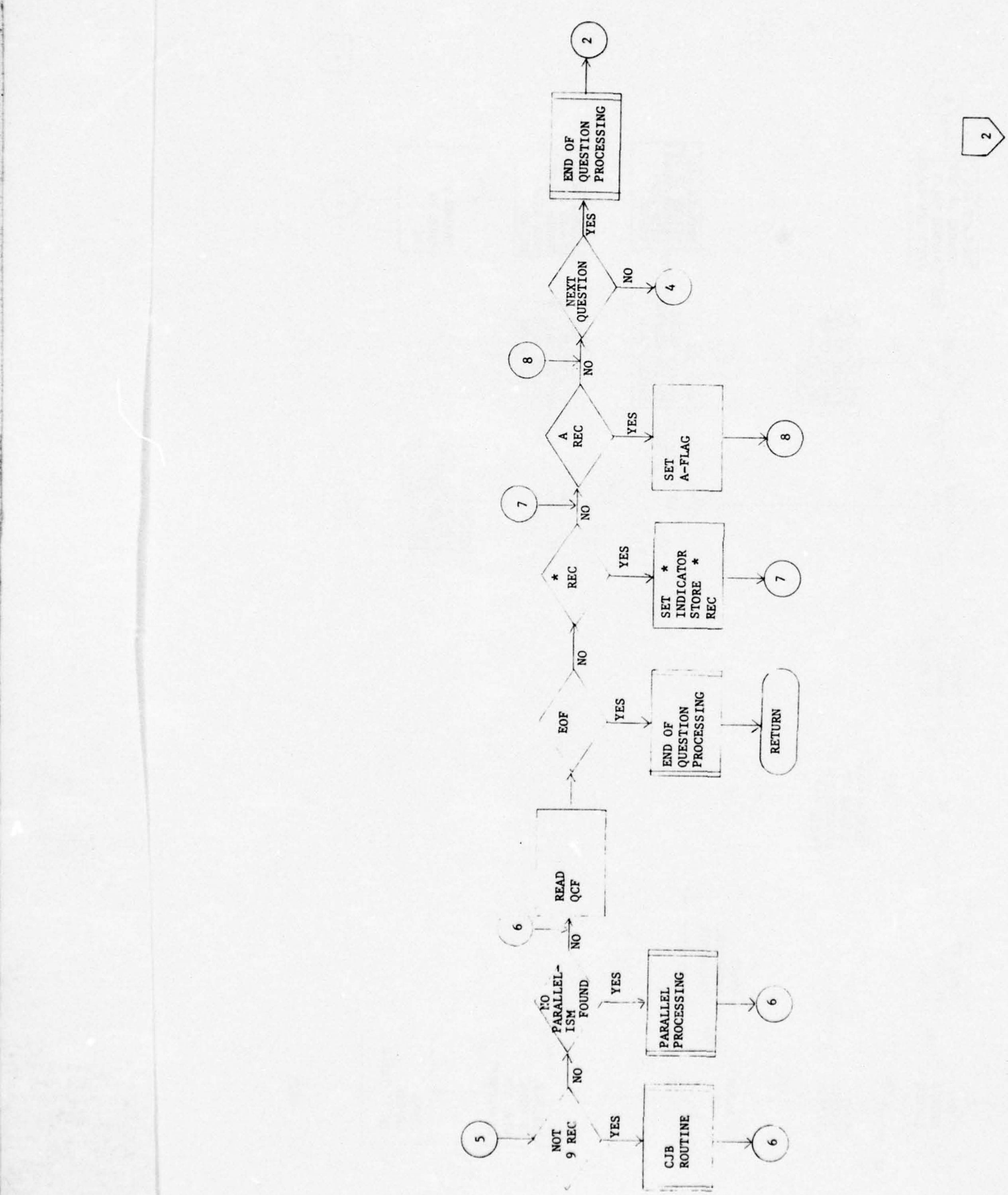
Next, SCRM will determine whether any 9 records were found. If some were found and parallelism exists, then all parallel indicators set by QTM for the L through Z constraints will be erased. Only those L-Z LIRCs which are also on the 9 record used for parallelism will have their parallel indicator set. If no parallelism was found, then SCRM will check the C and J LIRCs against all the L-Z LIRCs to see if parallelism can be established. If it is found, then the above procedure of erasing and then resetting the parallel indicators on L-Z records will occur. If it is not found, then all L-Z parallel indicators will be erased and nothing set. Next, C and J parallel indicators will be set for all questions that have parallelism.

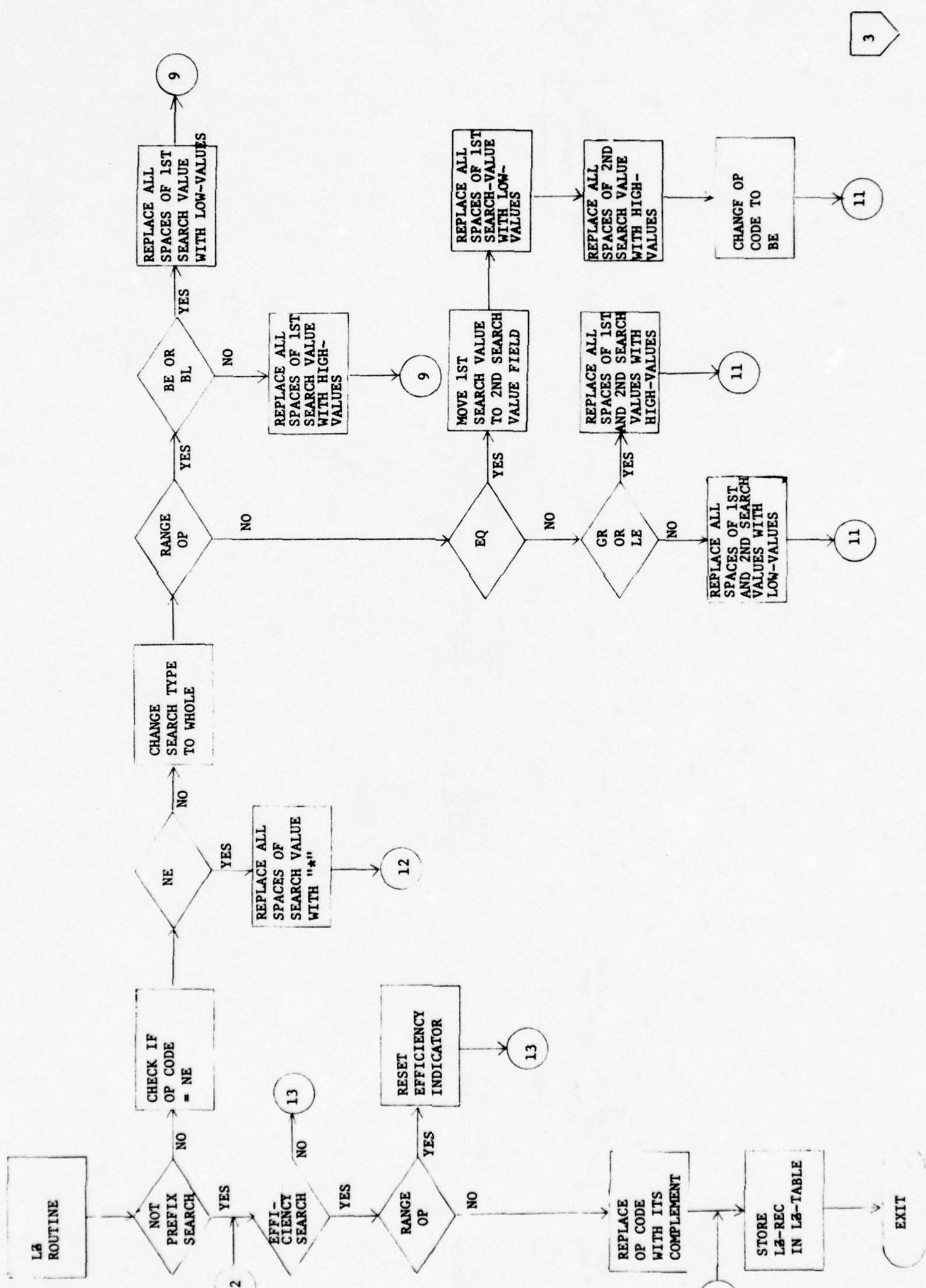
After the parallel processing, SCRM will finish the question by writing the stored records on the new QCF in the following order:

(1) L-Z records
(2) B record (if it exists)
(3) C, J records as encountered (or created)
(4) A, D, H records as encountered

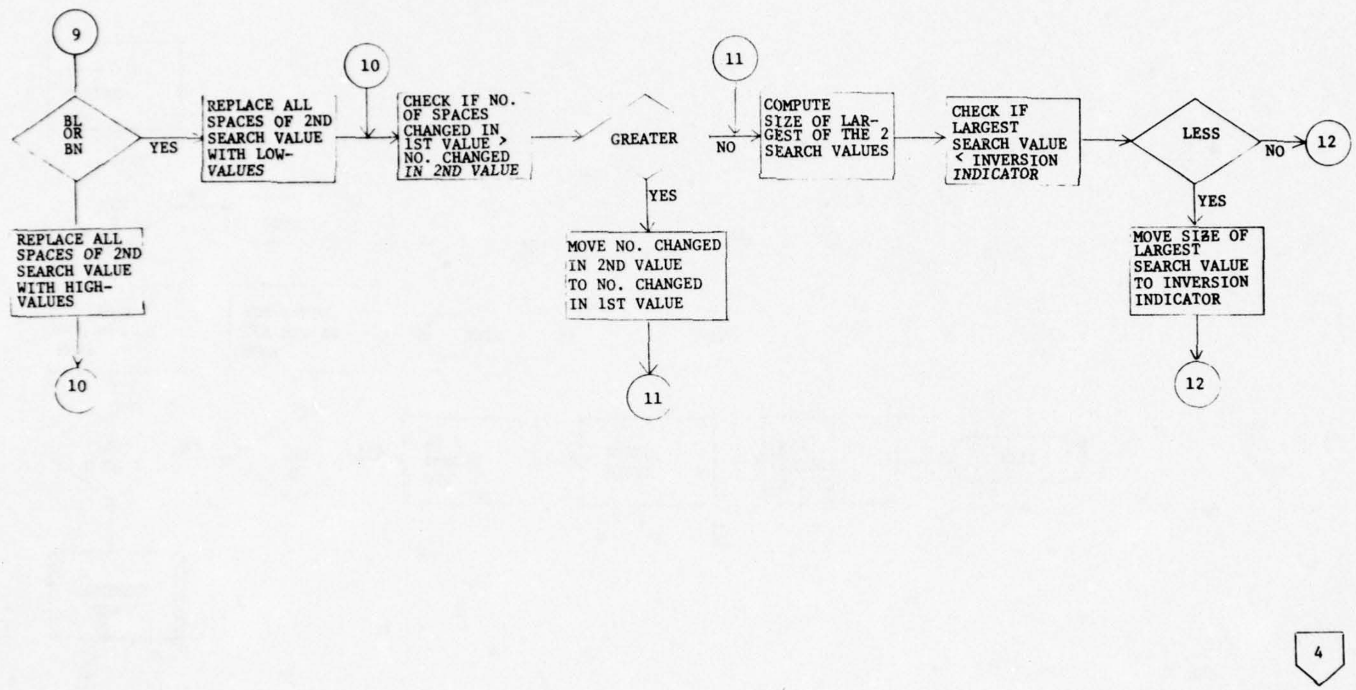
Type 9 records are not retained on the new QCF. Processing will continue with the next question.

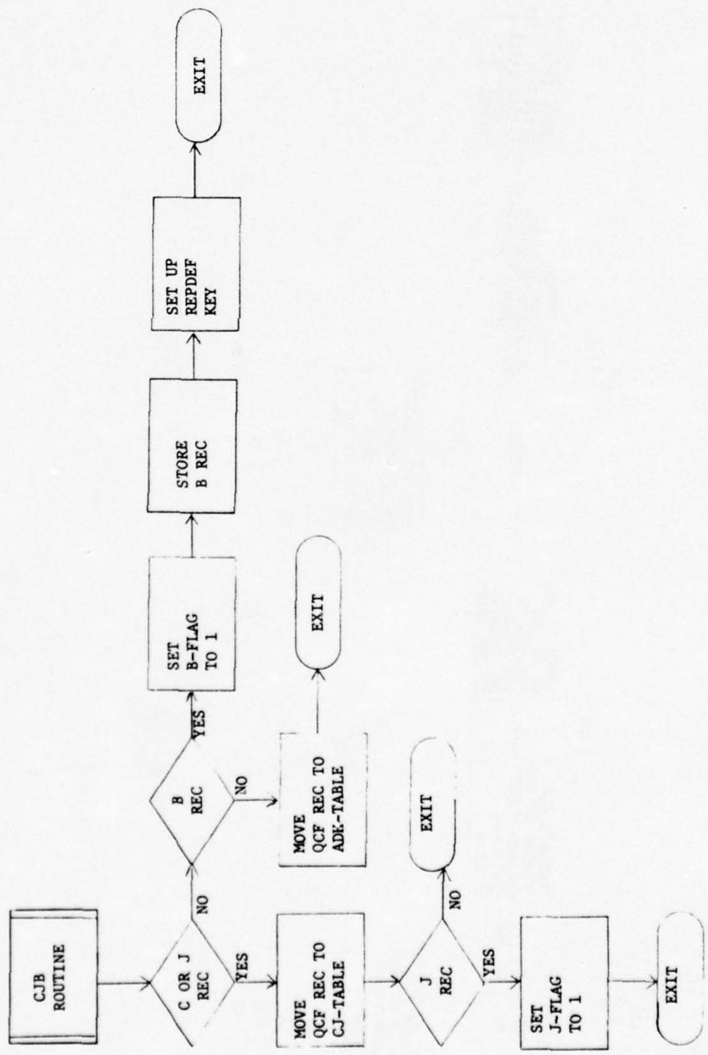

ENTRY SCRM
READ QCF
EOF
YES
NO
END OF QUESTION PROCESSING
RETURN
*
REC
SET * INDICATOR STORE * REC
1
A
REC
SET A-FLAG
2
*
REC
WRITE QCF REC
RESET * INDICATOR
READ QCF
EOF
RETURN
3
PRELIM. PROCESSING FOR NEW QUESTION
4
L-Z REC
L-Z ROUTINE
5
6
1


5
NOT 9 REC
YES
CJB ROUTINE
6
NO
NO PARALLEL-ISM FOUND
YES
PARALLEL PROCESSING
6
6
NO
READ QCF
EOF
YES
END OF QUESTION PROCESSING
RETURN
NO
* REC
YES
SET * INDICATOR STORE * REC
7
7
NO
A REC
YES
SET A-FLAG
8
8
NO
NEXT QUESTION
NO
4
YES
END OF QUESTION PROCESSING
2
2

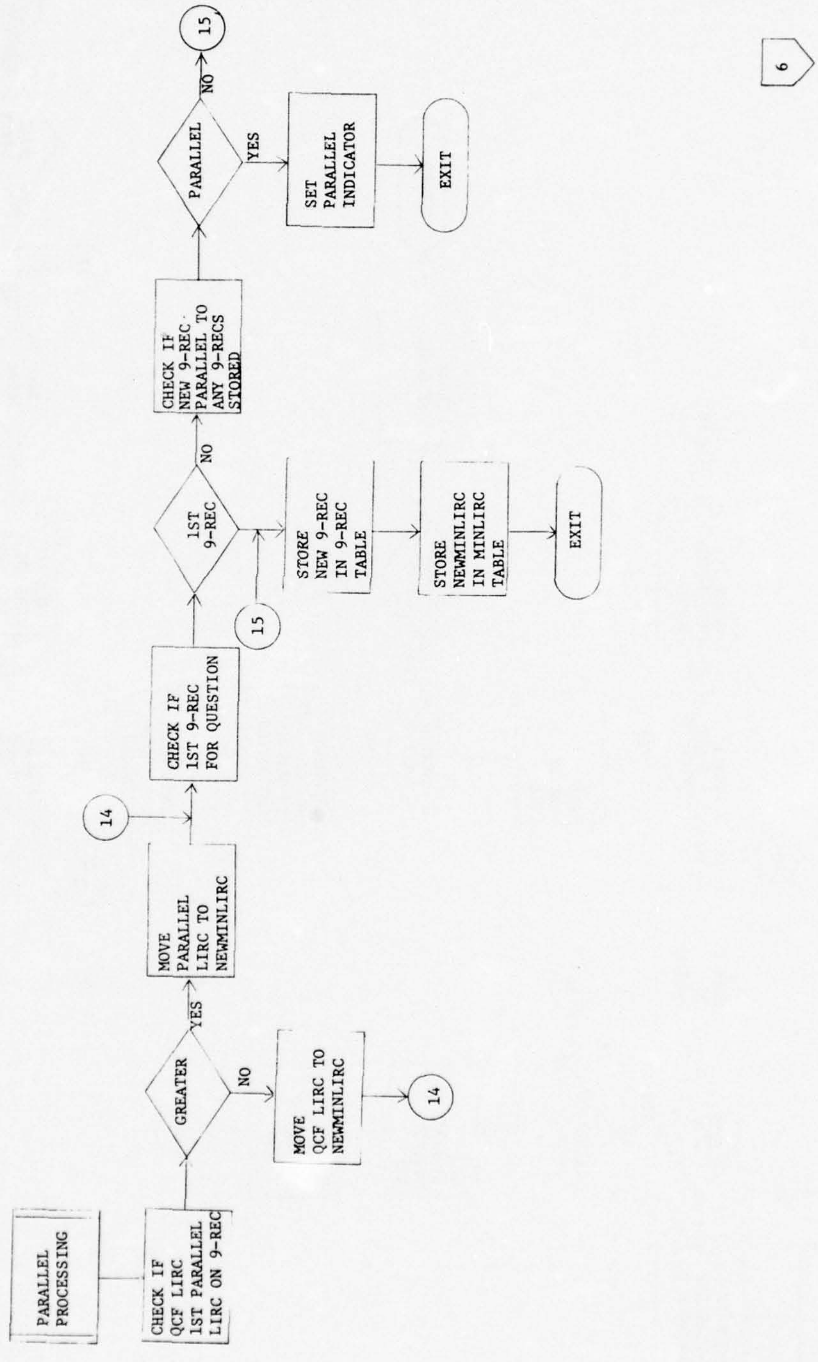


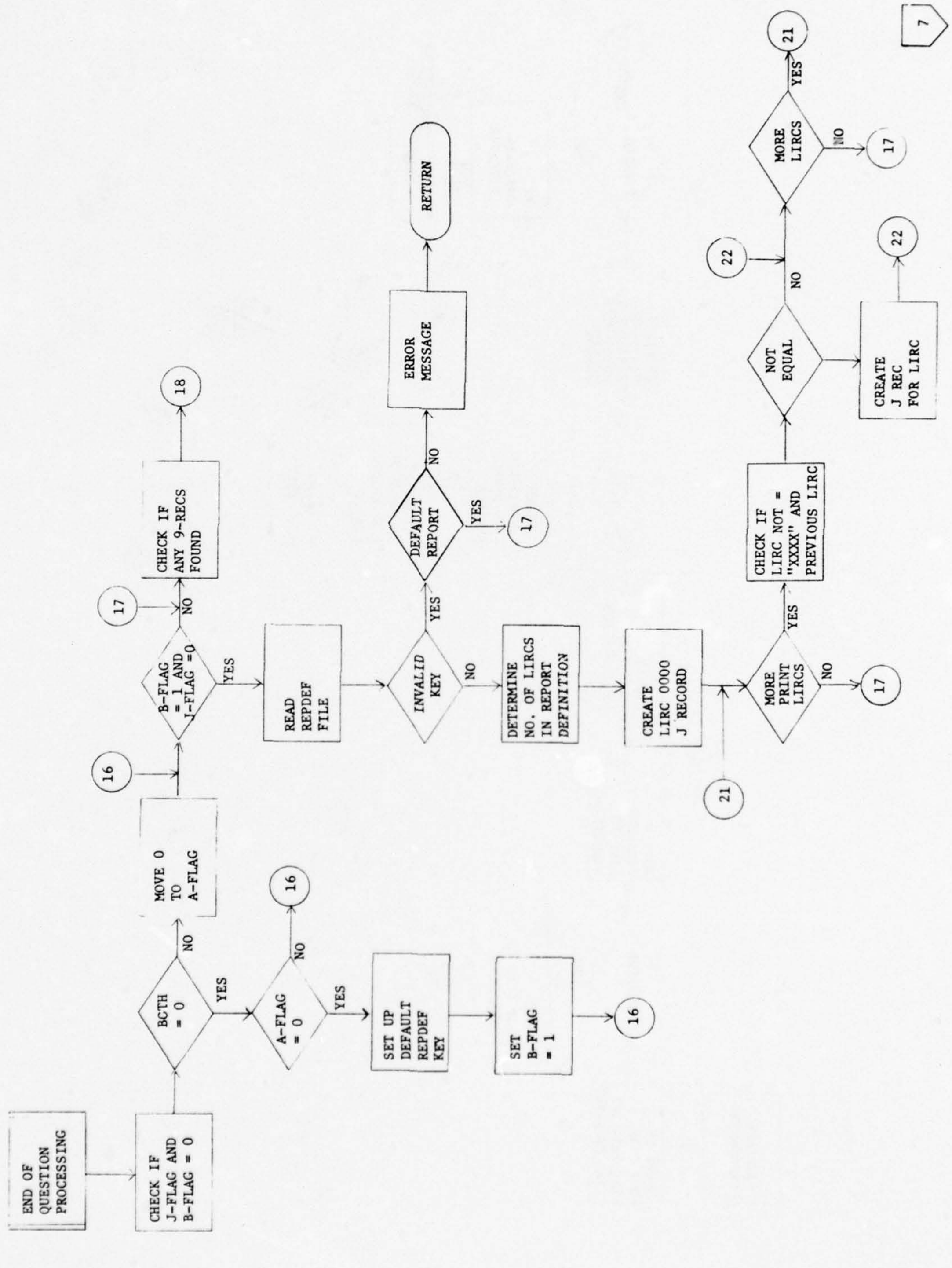
LƏ ROUTINE
NOT PREFIX SEARCH
YES
NO
CHECK IF OP CODE = NE
NE
YES
NO
REPLACE ALL SPACES OF SEARCH VALUE WITH "*"
12
CHANGE SEARCH TYPE TO WHOLE
RANGE OP
YES
NO
BE OR BL
YES
NO
REPLACE ALL SPACES OF 1ST SEARCH VALUE WITH LOW-VALUES
9
REPLACE ALL SPACES OF 1ST SEARCH VALUE WITH HIGH-VALUES
9
EQ
YES
NO
MOVE 1ST SEARCH VALUE TO 2ND SEARCH VALUE FIELD
REPLACE ALL SPACES OF 1ST SEARCH-VALUE WITH LOW-VALUES
REPLACE ALL SPACES OF 2ND SEARCH VALUE WITH HIGH-VALUES
CHANGF OP CODE TO BE
11
GR OR LE
YES
NO
REPLACE ALL SPACES OF 1ST AND 2ND SEARCH VALUES WITH HIGH-VALUES
11
REPLACE ALL SPACES OF 1ST AND 2ND SEARCH VALUES WITH LOW-VALUES
11
12
EFFI-CIENCY SEARCH
YES
NO
13
RANGE OP
YES
NO
RESET EFFICIENCY INDICATOR
13
REPLACE OP CODE WITH ITS COMPLEMENT
13
STORE LƏ-REC IN LƏ-TABLE
EXIT
3


9
BL OR BN
YES
REPLACE ALL SPACES OF 2ND SEARCH VALUE WITH LOW-VALUES
REPLACE ALL SPACES OF 2ND SEARCH VALUE WITH HIGH-VALUES
10
10
CHECK IF NO. OF SPACES CHANGED IN 1ST VALUE > NO. CHANGED IN 2ND VALUE
GREATER
YES
NO
MOVE NO. CHANGED IN 2ND VALUE TO NO. CHANGED IN 1ST VALUE
11
11
COMPUTE SIZE OF LARGEST OF THE 2 SEARCH VALUES
CHECK IF LARGEST SEARCH VALUE < INVERSION INDICATOR
LESS
NO
12
YES
MOVE SIZE OF LARGEST SEARCH VALUE TO INVERSION INDICATOR
12
4

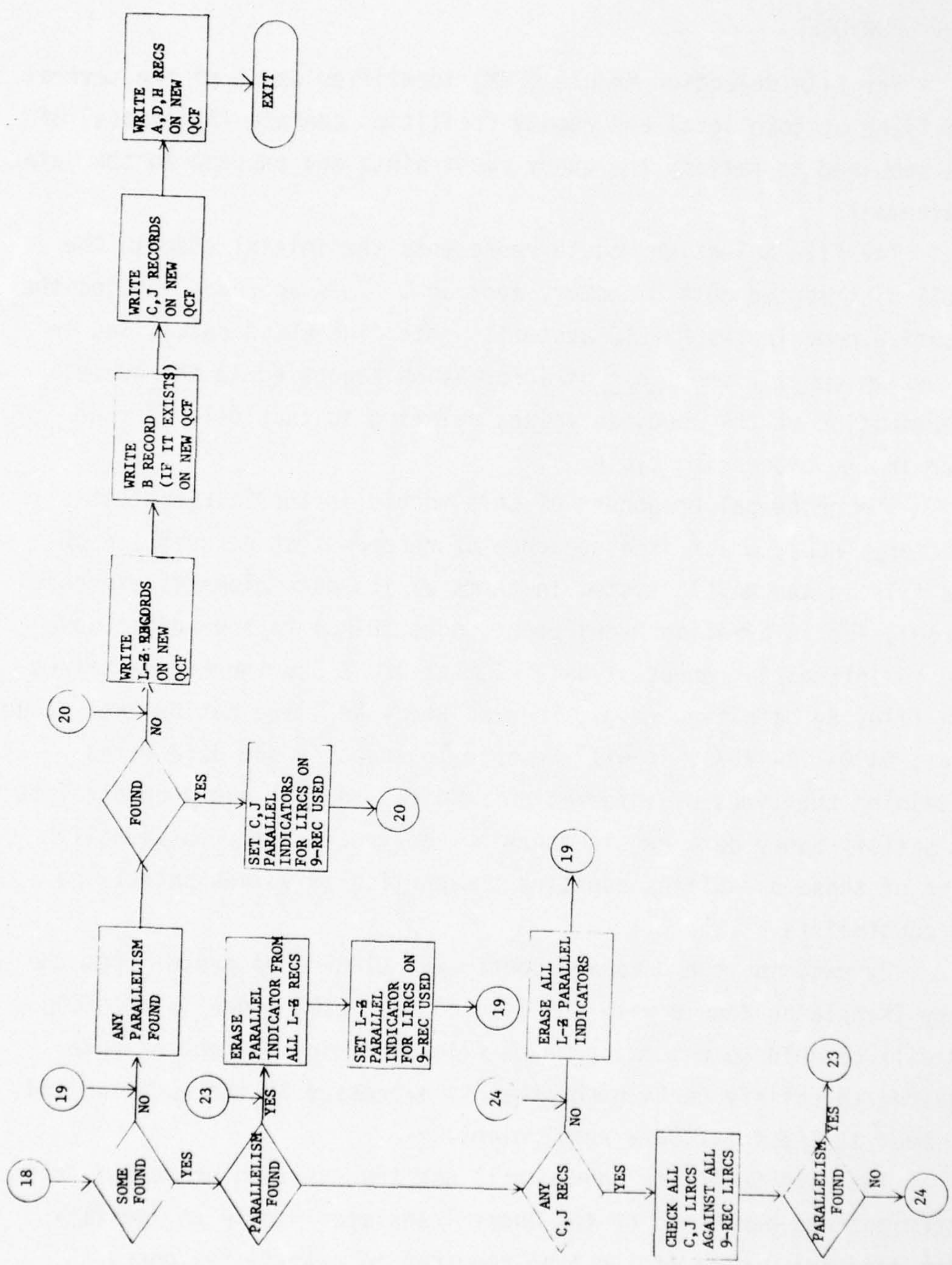

CJB ROUTINE
C OR J REC
YES
NO
MOVE QCF REC TO CJ-TABLE
J REC
YES
NO
SET J-FLAG TO 1
EXIT
EXIT
B REC
YES
NO
MOVE QCF REC TO ADK-TABLE
EXIT
SET B-FLAG TO 1
STORE B REC
SET UP REPDEF KEY
EXIT

5


PARALLEL PROCESSING
CHECK IF QCF LIRC 1ST PARALLEL LIRC ON 9-REC
GREATER
YES
NO
MOVE QCF LIRC TO NEWMINLIRC
14
MOVE PARALLEL LIRC TO NEWMINLIRC
14
CHECK IF 1ST 9-REC FOR QUESTION
1ST 9-REC
NO
15
STORE NEW 9-REC IN 9-REC TABLE
STORE NEWMINLIRC IN MINLIRC TABLE
EXIT
CHECK IF NEW 9-REC PARALLEL TO ANY 9-RECS STORED
PARALLEL
YES
NO
15
SET PARALLEL INDICATOR
EXIT
6



END OF QUESTION PROCESSING
CHECK IF J-FLAG AND B-FLAG = 0
BCTH = 0
YES
NO
MOVE 0 TO A-FLAG
16
A-FLAG = 0
NO
16
YES
SET UP DEFAULT REPDEF KEY
SET B-FLAG = 1
16
16
B-FLAG = 1 AND J-FLAG = 0
YES
NO
17
CHECK IF ANY 9-RECS FOUND
18
READ REPDEF FILE
INVALID KEY
YES
NO
DEFAULT REPORT
YES
17
NO
ERROR MESSAGE
RETURN
DETERMINE NO. OF LIRCS IN REPORT DEFINITION
CREATE LIRC 0000 J RECORD
21
MORE PRINT LIRCS
NO
17
YES
CHECK IF LIRC NOT = "XXXX" AND PREVIOUS LIRC
NOT EQUAL
CREATE J REC FOR LIRC
22
NO
22
MORE LIRCS
NO
17
YES
21
7



18
SOME FOUND
YES
NO
19
ANY PARALLELISM FOUND
FOUND
YES
NO
SET C,J PARALLEL INDICATORS FOR LIRCS ON 9-REC USED
20
20
WRITE L-Z RECORDS ON NEW QCF
WRITE B RECORD (IF IT EXISTS) ON NEW QCF
WRITE C,J RECORDS ON NEW QCF
WRITE A,D,H RECS ON NEW QCF
EXIT
PARALLELISM FOUND
YES
23
ERASE PARALLEL INDICATOR FROM ALL L-Z RECS
SET L-Z PARALLEL INDICATOR FOR LIRCS ON 9-REC USED
19
ANY C,J RECS
NO
24
ERASE ALL L-Z PARALLEL INDICATORS
19
YES
CHECK ALL C,J LIRCS AGAINST ALL 9-REC LIRCS
PARALLELISM FOUND
YES
23
NO
24

A.4 TITLE: FILE SELECTION MODULE (FSM)

A.4.1 PURPOSE

The File Selection Module (FSM) identifies which of the several data files at both local and remote facilities contain the type(s) of data required to satisfy the query constraints and respond to the data requirements.

The File Selection Module represents the initial step in the NAVLIS distributed data directory approach. This approach enables the Executive node in the NAVLIS system to determine which data files in the system contain the types of information requested in the query. Determination of the specific values relating to that file is made later in the processing cycle.

The principal component of this module is the File Content Directory (FCD), which is a sequence of records that describes each data file in the NAVLIS system in terms of its data elements expressed as Logistics Information Requirement Codes (LIRC) (a four-digit code used to internally represent data elements). Since the FCD describes data files by data type (e.g., Federal Stock No.) and not by data value (e.g., 5490-112-7767), it will be able to identify the data files containing the types of information required to meet query constraints and satisfy query data requirements but not necessarily to identify which of those data files contains the particular values satisfying the constraints.

By matching the data and constraint LIRC codes passed from the Query Translator Module with the file content LIRC codes in the FCD, FSM will be able to determine which files contain the type of data required to satisfy certain constraints expressed by the question and can meet at least one data requirement.

The File Selection Module will use the question/group/set logic determinations performed by the Query Translator Module to identify files containing data of the type required to satisfy the query. A data file will qualify for value searching if it contains at least one constraint type in each group of the question and at least one data

requirement expressed by each question. For queries with multiple questions, this procedure will be repeated for each question.

Additionally, the FSM is responsible for the only access verification performed by the Pilot Model (see Section 8.1.6 for discussion of security requirements). The NAVLIS Pilot Model restricted access at the file level through character identification code of users. Contained in the File Content Directory are "lock" records (one for each site in the network) associated with each data base in the network.

A.4.2 INPUTS

Inputs to the FSM consist of the following:

- Query records generated by the QTM with LIRC codes identifying the query constraints and data requirements, and
- The File Content Directory (FCD) identifying the LIRC coded contents of the systems data files. The entries in the FCD contain: for the content record,

    a) a file identification
    b) identification of the facility at which the file is located
    c) a series of LIRC codes, in numeric sequence, that correspond to the LIRC coded fields in the associated data file.

  for the lock record,

    a) a file identification
    b) a facility identification
    c) a series of user identifications at that facility (b) that have access to that file (a)

A.4.3 OUTPUTS

The FSM produces copies of the query constraint file generated by QTM for use in retrieving data from data files identified as candidates by FSM. For each candidate data file, a copy of the query constraint

file is written onto a single output file. That copy will always contain all the query constraints, but only those data selection records that request data contained on that specific data file. Each copy generated is preceded by a record identifying the file to which that copy applies. All copies for a particular site are grouped together and preceded by a record identifying that site.

When none of the NAVLIS data files can satisfy at least one of the constraints in each question group as well as at least one data requirement, the user is notified that the question cannot be satisfied.

A.4.4 RECORD FORMATS - (See Figures A1-A5 and A9)

A.4.5 STORAGE REQUIREMENTS

Program (less library modules) occupies 20,000 bytes on UNIVAC Series 70/45.

A.4.6 PROCESSING DESCRIPTION (Refer to FSM Flowchart, page 70)

The FSM is called by the Executive Control Module (ECM) after the Query Translation Module (QTM) has successfully completed processing. The FSM commences by allowing the user to input file parameters or descriptors that describe the kinds of data bases of interest. The user may be interested in a specific subject (e.g., aircraft maintenance), or organization (e.g., COMNAVAIRPAC), or location (e.g., West Coast), etc. The FSM maintains a parameter table associating each of the data bases in the system with a list of parameters describing the contents of the data base and other significant data base characteristics. When the user elects to input file parameters, they are compared with the contents of the file parameter table, and the identification(s) of the data base(s) described by the user parameters are stored in an eligibility table. The user may also elect not to specify file parameters, in which case all data bases in the system are eligible for further processing. The Query Constraint File (QCF) produced by QTM is then read

Section & Page 101.

Subsection

Date Documented

Change Notice #

Fold back at dotted line.

CARD OR RECORD LAYOUT - DOUBLE

SYSTEM ID: ☐ New ☐ Revision

RECORD TITLE: LOCK RECORD

FILE ID: FILE CONTENT DIRECTORY

RECORD LENGTH: 212 MINIMUM ☐ FIXED ☑ MAX.

WORDS ☐ BYTES ☐ CHARACTERS ☑

BLOCKING FACTOR 1 RECORD/BLOCK

MEDIUM:

CARD ☐ TAPE ☐ DASD ☐

OTHER ☐

DENSITY

TRACKS: 7 ☐ 9 ☐

CARD STOCK

CHARACTER ENCODING:

HOL ☐ EXH ☐

BCD ☐ EBCDIC ☐ ASCII ☐

PACKED DECIMAL ☐

BINARY OR NON-CHARACTER INFORMATION ☐

OTHER ☐

PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:

TRAILER:

COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Positions 1	2–3	4–9	10–100
REC TYPE	FILE ID	SITE ID	REPEATING 3 CHAR USER ID'S
(1)	(2)	(3)	(4)

(1) "L"
(2) ALPHANUMERIC
(3) ALPHABETIC
(4) ALPHABETIC

RECORD TITLE: CONTENT RECORD

FILE ID: FILE CONTENT DIRECTORY

RECORD LENGTH: 212 MINIMUM ☐ FIXED ☑ MAX.

WORDS ☐ BYTES ☐ CHARACTERS ☑

BLOCKING FACTOR 1 RECORD/BLOCK

MEDIUM:

CARD ☐ TAPE ☐ DASD ☐

OTHER ☐

DENSITY

TRACKS: 7 ☐ 9 ☐

CARD STOCK

CHARACTER ENCODING:

HOL ☐ EXH ☐

BCD ☐ EBCDIC ☐ ASCII ☐

PACKED DECIMAL ☐

BINARY OR NON-CHARACTER INFORMATION ☐

OTHER ☐

PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:

TRAILER:

COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Positions 1	2–3	4–9	10	11–100
REC TYPE	FILE ID	SITE ID	FILE TYPE	REPEATING 4 CHAR LIRC'S
(1)	(2)	(3)	(4)	(5)

(1) "F"
(2) ALPHANUMERIC
(3) ALPHABETIC
(4) "R" OR "S"
(5) NUMERIC

Figure A-9

into an internal table by FSM to facilitate rapid execution of multiple reads of the file. Pilot Model constraints on the size of the QCF made this approach feasible. The larger size of the QCF in an operational mode would necessitate an alternative method.

The following sequence of operations is repeated for each data base identified in the eligibility table, if the user specified file parameters at the outset, or for each data base in the system, if no file parameters were specified. For each data base in the system, the FCD contains one or more lock records for each site in the network identifying, by user identification, those users with access to that data base. The lock records are followed by one or more content records that indicate by LIRC the types of data contained in that data base. The FSM reads the first series of lock records in the FCD, locating the record whose site identification corresponds to the site identification of the user passed to the FSM by the ECM. The user identifications contained in that record are compared to the user identification also passed by the ECM. Finding the user identification in the lock record indicates this user is authorized to retrieve data from the associated data base. The lock record may also contain the value "ALL" instead of a series of user identifications, indicating that no access restrictions are associated with this data base. If neither the user identification nor the value "ALL" is found in the lock record, the FCD is read until the series of lock records for the next data base is encountered and the access validation process begins for that data base. When the user's identification or the value "ALL" is encountered in a lock record for the user's site, the remaining lock records, if any, are skipped until the content record for that data base is encountered.

The query records generated by QTM are then accessed and the LIRC codes in the query records for the first constraint group (Group Field L) are compared to the LIRC codes in the FCD content record. If at least one match is found, the LIRC codes in the query records for the next constraint group (Group M, if the query contains multiple constraint groups) are compared to the same FCD entry. This process is continued

until

- A match has been found between the LIRC codes in the FCD content record being processed and at least one constraint LIRC in each constraint group of the question, or

- All LIRC codes have been searched and no match has been found between the LIRC codes in a constraint group and those in the FCD content record.

This latter situation indicates that the data file represented by the FCD entry currently being processed does not contain the data required to satisfy at least one constraint in each constraint group of the question. The FSM then reads to the next series of lock records in the FCD to begin the access validation processing for the next data base.

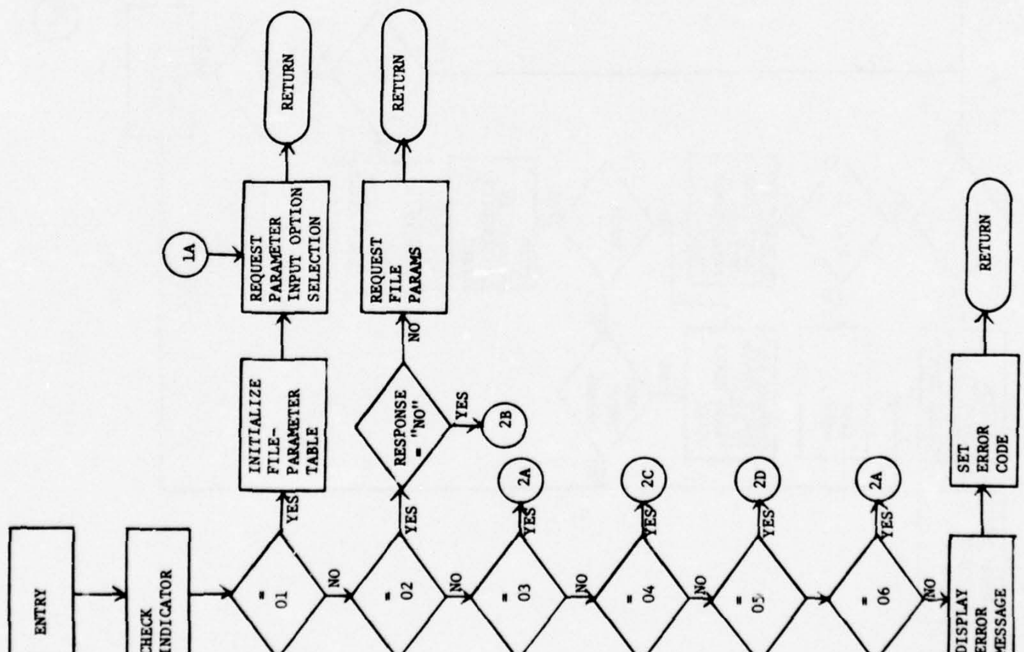
When a match is found, the FSM must determine whether the data file represented by the FCD being processed satisfies any of the data requirements expressed by the question, as well as meeting the necessary constraint conditions. The data requirements for the query are expressed in the query records with a Group field of "J", "C", or "D". The FSM retrieves the LIRC code from each J, C, or D group query record and compares it to the LIRC codes in the FCD entry being processed. If none of the LIRC codes identifying data requirements of the questions being processed are found in the FCD entry, that file does not satisfy any data requirements, even though the data file represented by that FCD entry satisfies the question constraint conditions. The FSM then reads to the next series of lock records in the FCD and begins the access validation and content comparison for the next data base.

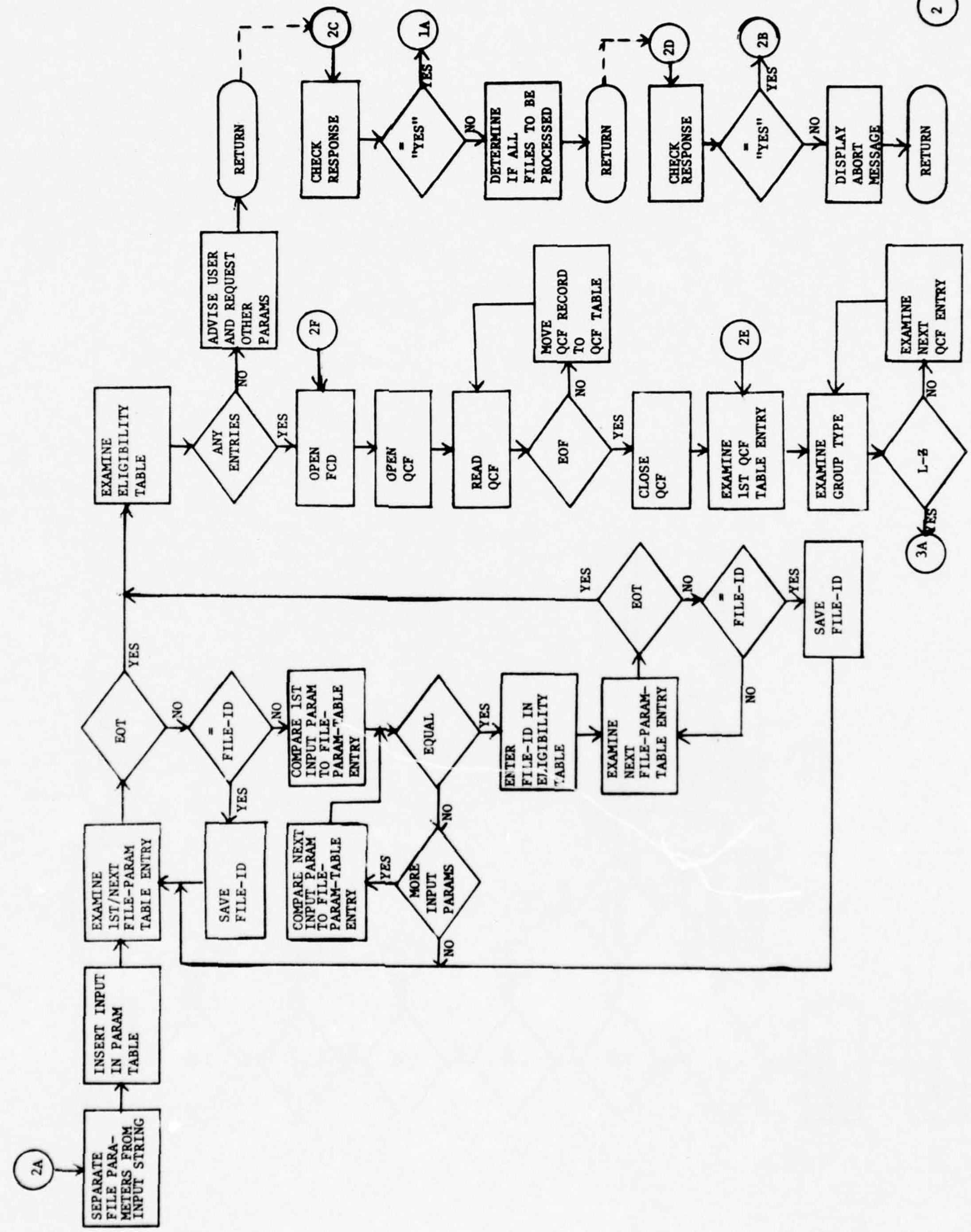
When the FSM finds the first match between a data requirement (Group J, or C or D if there are no J-records) LIRC code and a LIRC code in the FCD entry, the module enters the identification and site location of the data base represented by the content record into a Candidate File Table (CFT). The entries in this table will eventually represent all those data bases in the NAVLIS system that satisfy the query constraints and at least one data retrieval requirement. The LIRC codes of each of the data requirements satisfied are placed

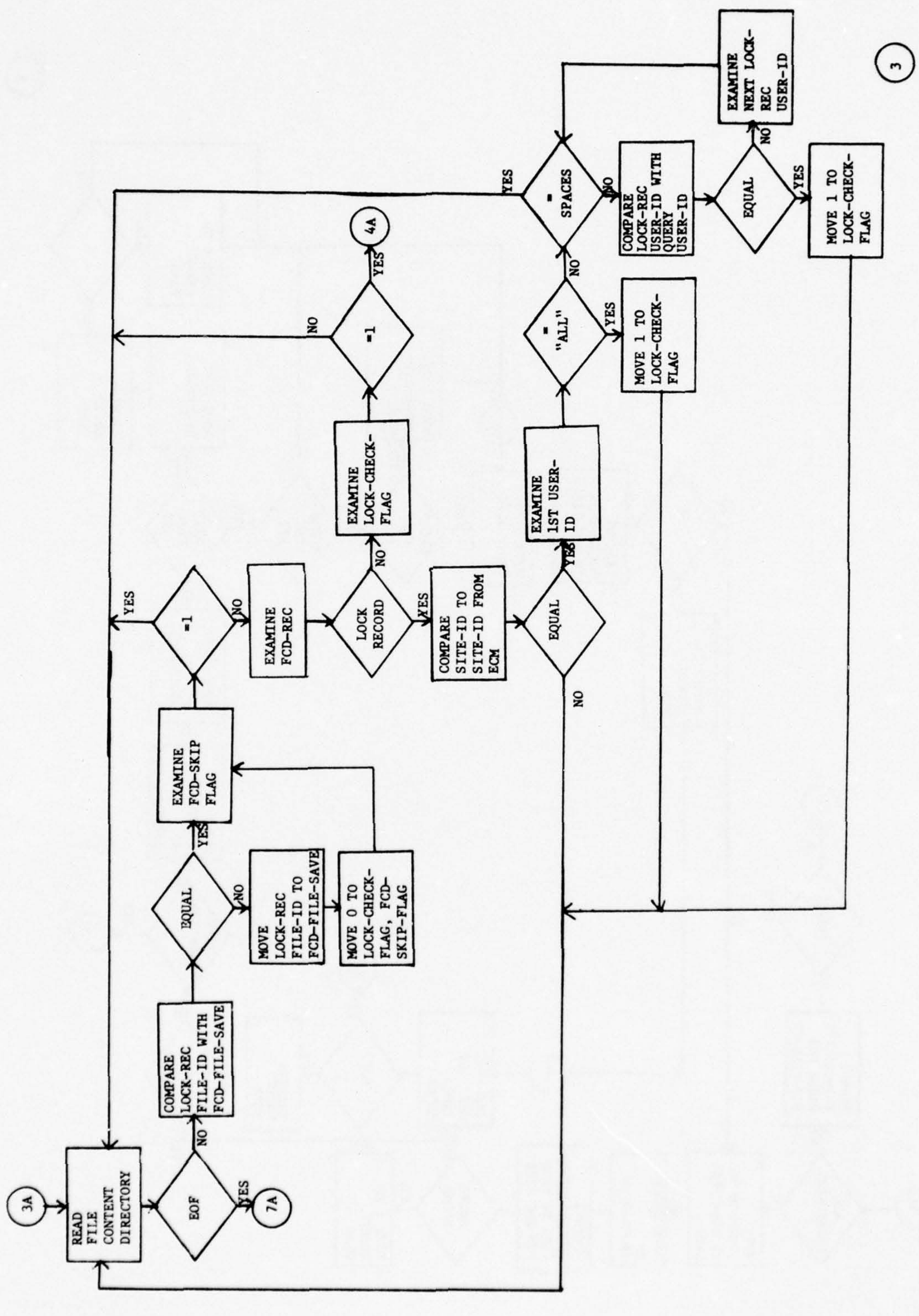
sequentially in an associated table, the LIRC-Table. The LIRC-Table indices indicating the first and last LIRC code applying to the data base being processed are entered in the Candidate File Table along with the data base identification and location.

The process described above for access validation, constraint satisfaction, and data requirement satisfaction is repeated for each data base in theEl igibility Table or in the FCD. When the FCD records have been exhausted, the FSM verifies that at least one entry has been made in the Candidate File Table; if not, the user is notified that none of the data bases in the system contain the types of data required to respond to his query. When entries have been made in the CFT, the FSM generates an output file containing a copy of the QCF for each candidate data base. The first record in the output file contains the site identification and file identification of the first candidate data base. Each of the constraint records on the QCF and the data requirement records satisfied by the data base are duplicated. Copies of the QCF for data bases at any single site are contiguous with the first record for that location containing a site identification field. That field is used by ECM to determine the location in the network to which the following copies of the QCF are to be transmitted. The first record of each QCF copy contains an identification of the data base to which that copy applies.

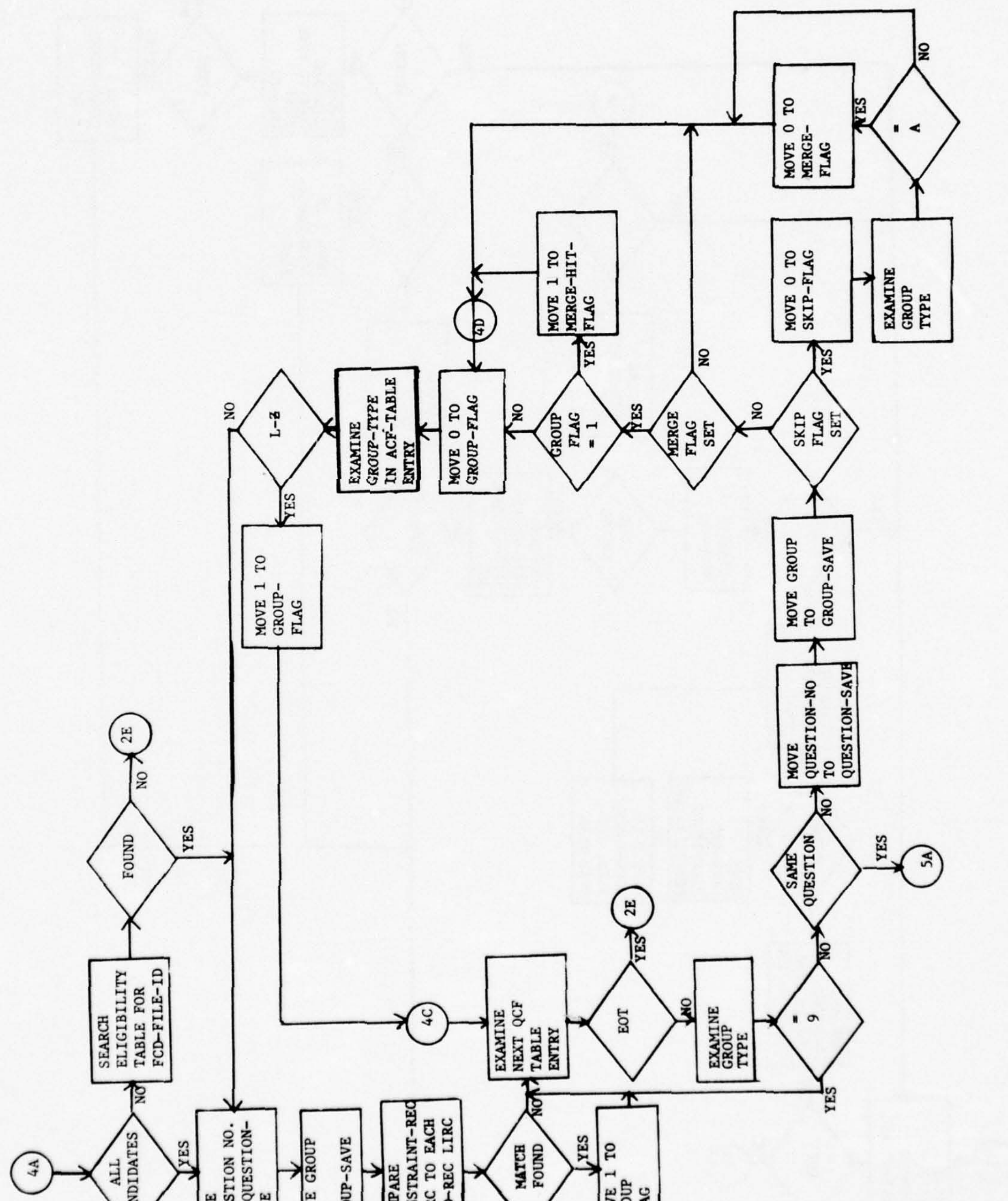
1

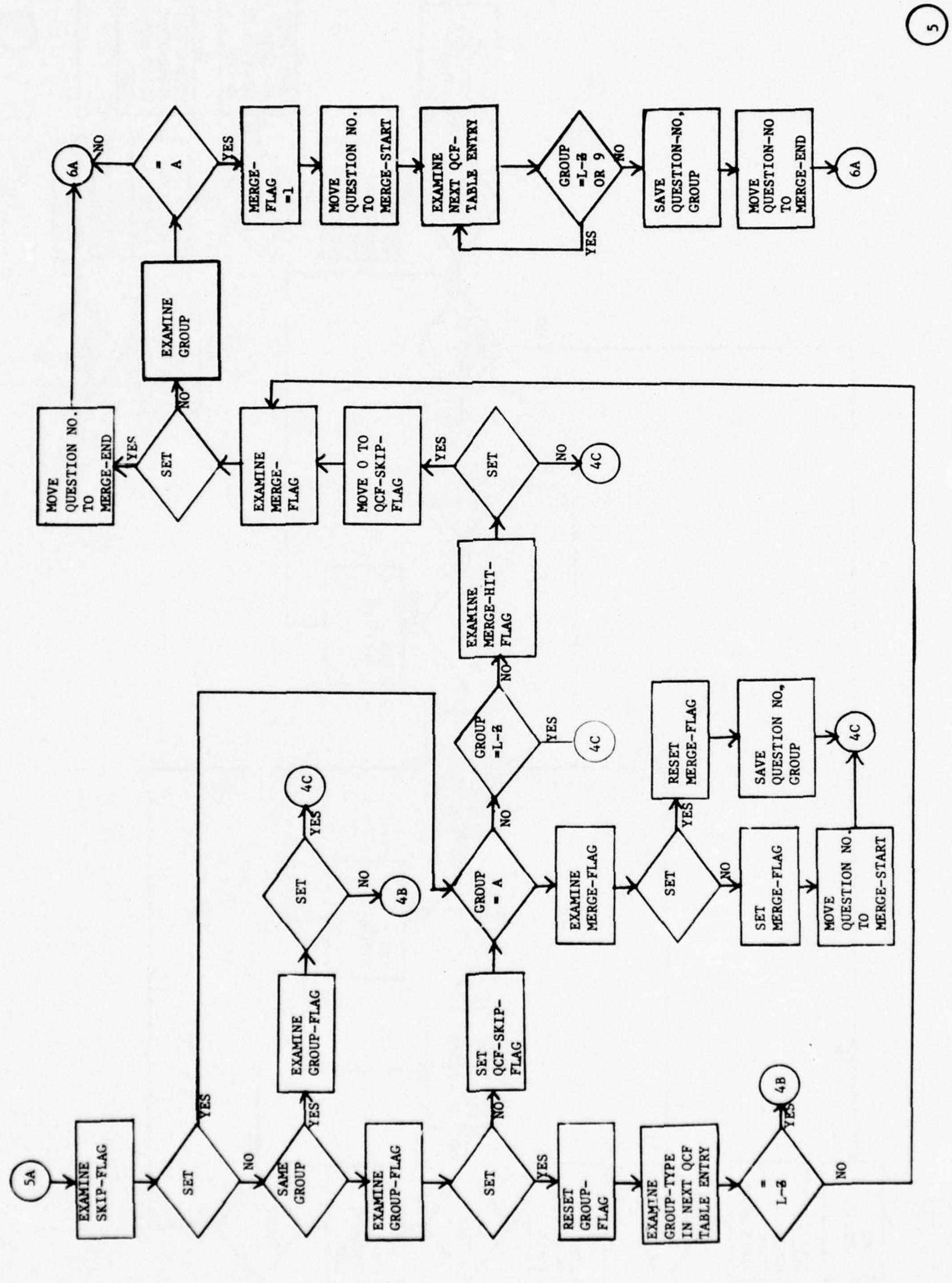

ENTRY
CHECK INDICATOR
= 01
YES
NO
INITIALIZE FILE-PARAMETER TABLE
1A
REQUEST PARAMETER INPUT OPTION SELECTION
RETURN
= 02
RESPONSE = "NO"
YES
2B
NO
REQUEST FILE PARAMS
RETURN
= 03
2A
= 04
2C
= 05
2D
= 06
2A
DISPLAY ERROR MESSAGE
SET ERROR CODE
RETURN

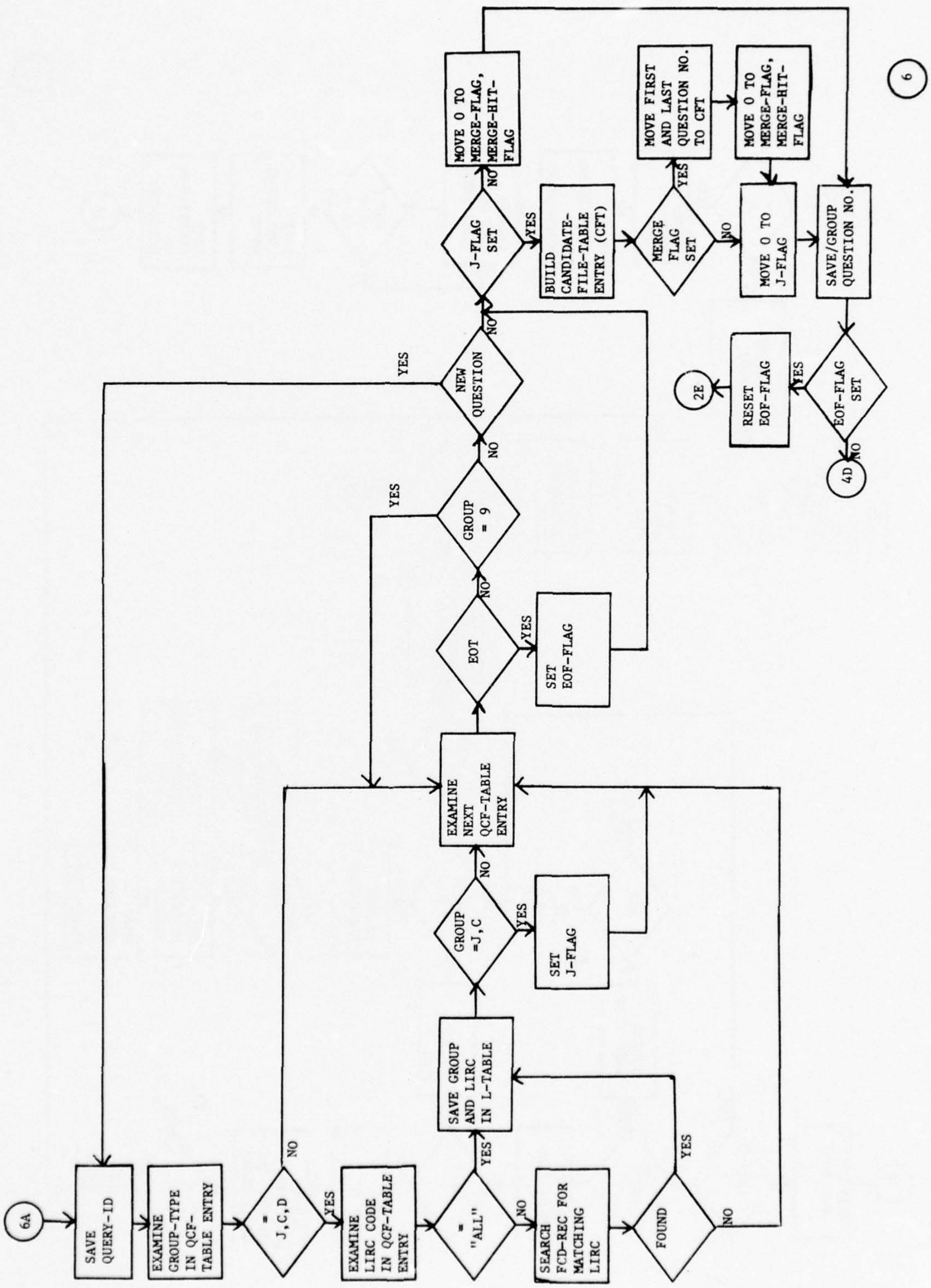

2A
SEPARATE FILE PARA- METERS FROM INPUT STRING
INSERT INPUT IN PARAM TABLE
EXAMINE 1ST/NEXT FILE-PARAM TABLE ENTRY
EOT
YES
NO
= FILE-ID
YES
NO
SAVE FILE-ID
COMPARE 1ST INPUT PARAM TO FILE- PARAM-TABLE ENTRY
EQUAL
YES
NO
MORE INPUT PARAMS
YES
NO
COMPARE NEXT INPUT PARAM TO FILE- PARAM-TABLE ENTRY
ENTER FILE-ID IN ELIGIBILITY TABLE
EXAMINE NEXT FILE-PARAM- TABLE ENTRY
EOT
YES
NO
= FILE-ID
YES
NO
SAVE FILE-ID
EXAMINE ELIGIBILITY TABLE
ANY ENTRIES
YES
NO
ADVISE USER AND REQUEST OTHER PARAMS
RETURN
2C
CHECK RESPONSE
= "YES"
YES
1A
NO
DETERMINE IF ALL FILES TO BE PROCESSED
RETURN
2D
CHECK RESPONSE
= "YES"
YES
2B
NO
DISPLAY ABORT MESSAGE
RETURN
2F
OPEN FCD
OPEN QCF
READ QCF
EOF
NO
MOVE QCF RECORD TO QCF TABLE
YES
CLOSE QCF
2E
EXAMINE 1ST QCF TABLE ENTRY
EXAMINE GROUP TYPE
L-B
NO
EXAMINE NEXT QCF ENTRY
YES
3A
2


3A
READ FILE CONTENT DIRECTORY
EOF
YES
7A
NO
COMPARE LOCK-REC FILE-ID WITH FCD-FILE-SAVE
EQUAL
NO
MOVE LOCK-REC FILE-ID TO FCD-FILE-SAVE
MOVE 0 TO LOCK-CHECK-FLAG, FCD-SKIP-FLAG
YES
EXAMINE FCD-SKIP FLAG
=1
YES
NO
EXAMINE FCD-REC
LOCK RECORD
NO
EXAMINE LOCK-CHECK-FLAG
=1
NO
YES
4A
YES
COMPARE SITE-ID TO SITE-ID FROM ECM
EQUAL
NO
YES
EXAMINE 1ST USER-ID
= "ALL"
YES
MOVE 1 TO LOCK-CHECK-FLAG
NO
= SPACES
YES
NO
COMPARE LOCK-REC USER-ID WITH QUERY USER-ID
EQUAL
NO
EXAMINE NEXT LOCK-REC USER-ID
YES
MOVE 1 TO LOCK-CHECK-FLAG
3

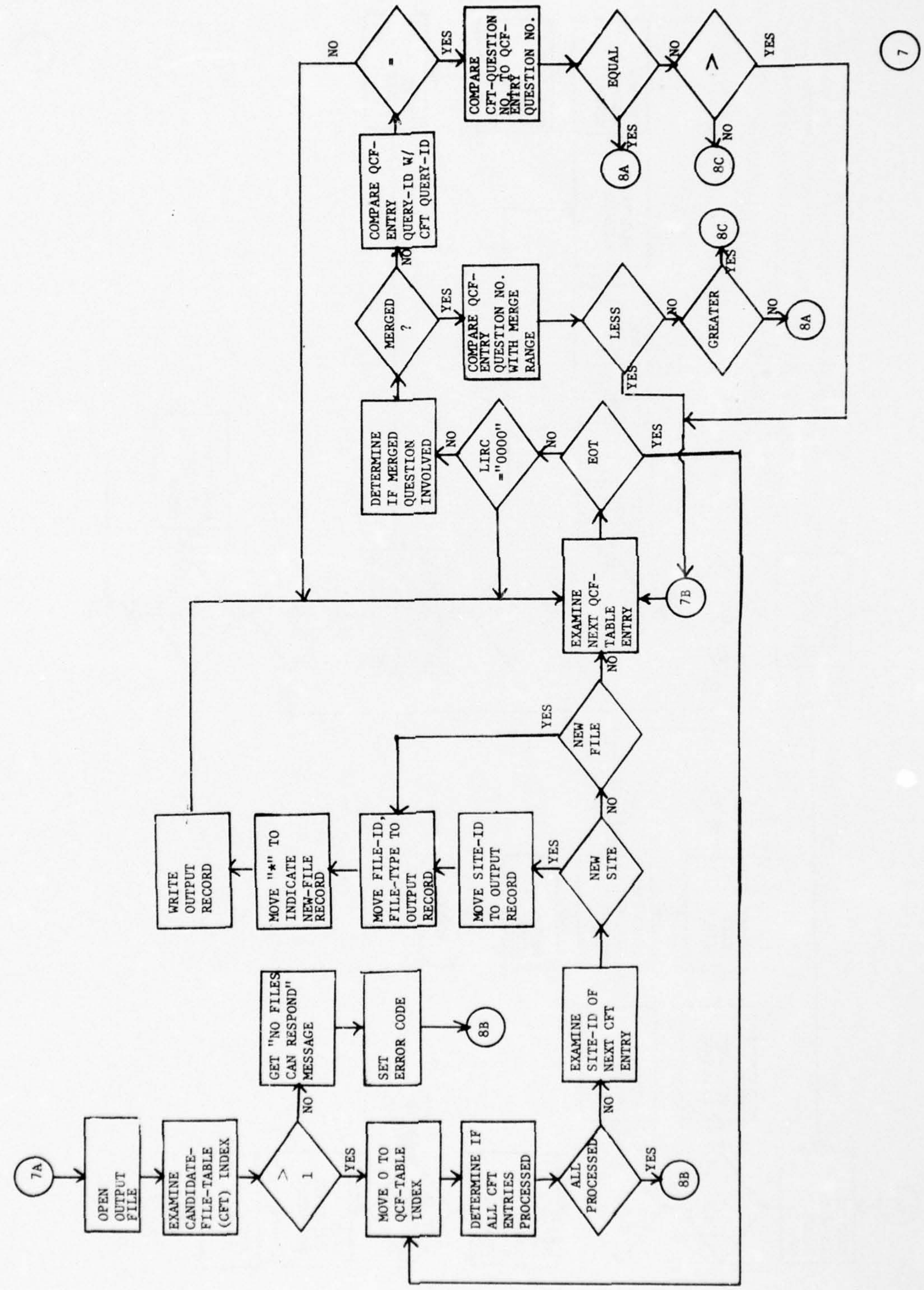
4


4A
ALL CANDIDATES
YES
NO
SEARCH ELIGIBILITY TABLE FOR FCD-FILE-ID
FOUND
YES
NO
2E
MOVE QUESTION NO. TO QUESTION-SAVE
MOVE GROUP TO GROUP-SAVE
COMPARE CONSTRAINT-REC LIRC TO EACH FCD-REC LIRC
MATCH FOUND
YES
NO
MOVE 1 TO GROUP FLAG
4C
EXAMINE NEXT QCF TABLE ENTRY
EOT
YES
NO
2E
EXAMINE GROUP TYPE
= 9
YES
NO
SAME QUESTION
YES
NO
5A
MOVE QUESTION-NO TO QUESTION-SAVE
MOVE GROUP TO GROUP-SAVE
SKIP FLAG SET
YES
NO
MOVE 0 TO SKIP-FLAG
EXAMINE GROUP TYPE
= A
YES
NO
MOVE 0 TO MERGE-FLAG
MERGE FLAG SET
YES
NO
GROUP FLAG = 1
YES
NO
MOVE 1 TO MERGE-HIT-FLAG
4D
MOVE 0 TO GROUP-FLAG
EXAMINE GROUP-TYPE IN ACF-TABLE ENTRY
L-Z
YES
NO
MOVE 1 TO GROUP-FLAG

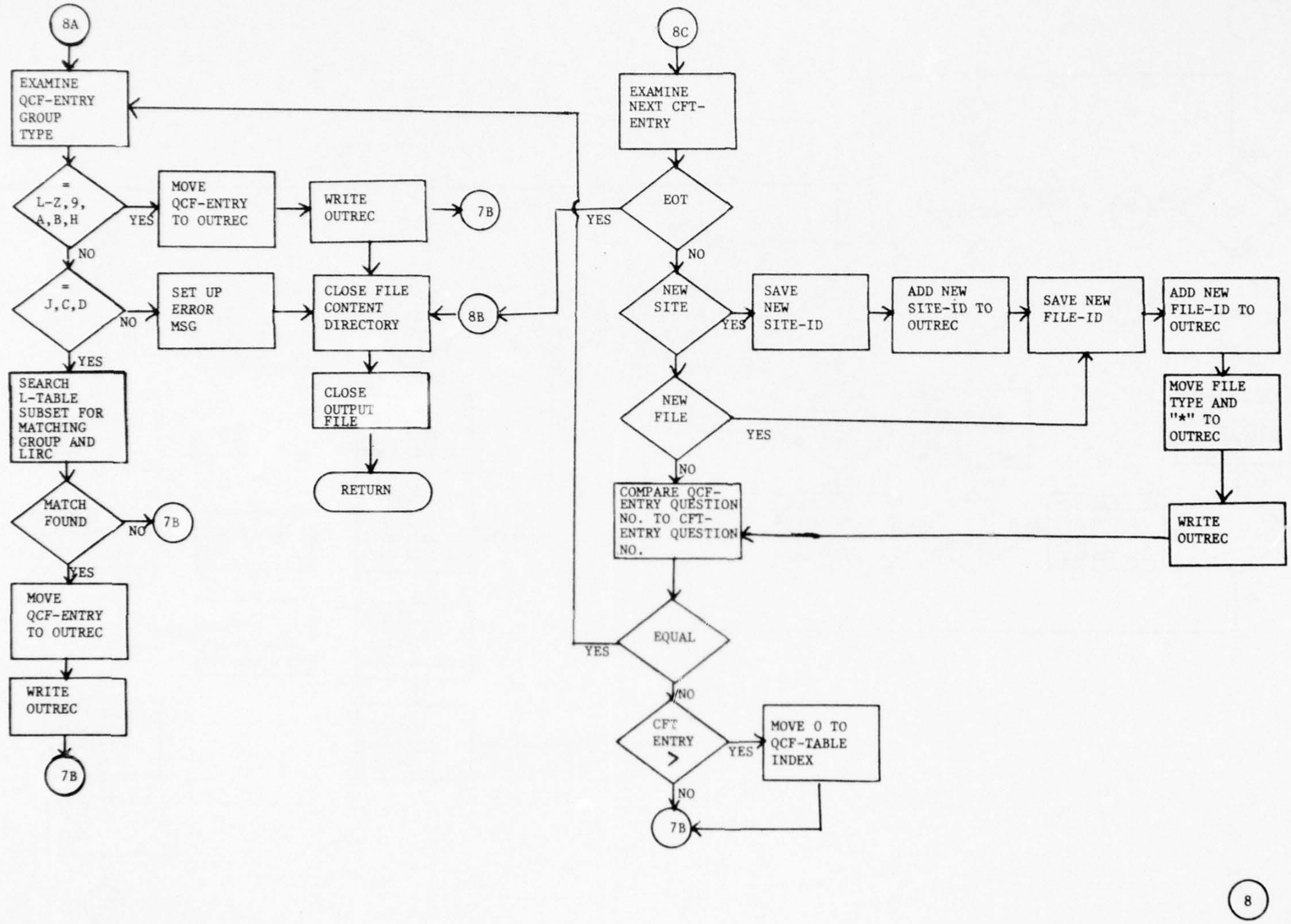

5A
EXAMINE SKIP-FLAG
SET
YES
NO
SAME GROUP
YES
EXAMINE GROUP-FLAG
SET
YES
4C
NO
4B
EXAMINE GROUP-FLAG
SET
NO
SET QCF-SKIP-FLAG
YES
RESET GROUP-FLAG
EXAMINE GROUP-TYPE IN NEXT QCF TABLE ENTRY
= L-Z
YES
4B
NO
GROUP = A
NO
GROUP =L-Z
YES
4C
NO
EXAMINE MERGE-HIT-FLAG
SET
NO
4C
YES
MOVE 0 TO QCF-SKIP-FLAG
EXAMINE MERGE-FLAG
SET
YES
MOVE QUESTION NO. TO MERGE-END
NO
EXAMINE GROUP
6A
= A
NO
YES
MERGE-FLAG =1
MOVE QUESTION NO. TO MERGE-START
EXAMINE NEXT QCF-TABLE ENTRY
GROUP =L-Z OR 9
YES
NO
SAVE QUESTION-NO, GROUP
MOVE QUESTION-NO TO MERGE-END
6A
EXAMINE MERGE-FLAG
SET
YES
RESET MERGE-FLAG
SAVE QUESTION NO, GROUP
4C
NO
SET MERGE-FLAG
MOVE QUESTION NO. TO MERGE-START
5



6A
SAVE QUERY-ID
EXAMINE GROUP-TYPE IN QCF-TABLE ENTRY
= J,C,D
YES
NO
EXAMINE LIRC CODE IN QCF-TABLE ENTRY
= "ALL"
YES
NO
SAVE GROUP AND LIRC IN L-TABLE
SEARCH FCD-REC FOR MATCHING LIRC
FOUND
YES
NO
GROUP =J,C
YES
NO
SET J-FLAG
EXAMINE NEXT QCF-TABLE ENTRY
EOT
YES
NO
SET EOF-FLAG
GROUP = 9
YES
NO
NEW QUESTION
YES
NO
J-FLAG SET
YES
NO
MOVE 0 TO MERGE-FLAG, MERGE-HIT-FLAG
BUILD CANDIDATE-FILE-TABLE ENTRY (CFT)
MERGE FLAG SET
YES
NO
MOVE FIRST AND LAST QUESTION NO. TO CFT
MOVE 0 TO MERGE-FLAG, MERGE-HIT-FLAG
MOVE 0 TO J-FLAG
SAVE/GROUP QUESTION NO.
EOF-FLAG SET
YES
NO
RESET EOF-FLAG
2E
4D
6



7A
OPEN OUTPUT FILE
EXAMINE CANDIDATE-FILE-TABLE (CFT) INDEX
> 1
YES
NO
GET "NO FILES CAN RESPOND" MESSAGE
SET ERROR CODE
8B
MOVE 0 TO QCF-TABLE INDEX
DETERMINE IF ALL CFT ENTRIES PROCESSED
ALL PROCESSED
YES
8B
NO
EXAMINE SITE-ID OF NEXT CFT ENTRY
NEW SITE
YES
MOVE SITE-ID TO OUTPUT RECORD
MOVE FILE-ID, FILE-TYPE TO OUTPUT RECORD
MOVE "*" TO INDICATE NEW-FILE RECORD
WRITE OUTPUT RECORD
NO
NEW FILE
YES
NO
EXAMINE NEXT QCF-TABLE ENTRY
7B
EOT
YES
NO
LIRC ="0000"
NO
DETERMINE IF MERGED QUESTION INVOLVED
MERGED ?
YES
COMPARE QCF-ENTRY QUESTION NO. WITH MERGE RANGE
LESS
YES
NO
GREATER
YES
8C
NO
8A
NO
COMPARE QCF-ENTRY QUERY-ID W/ CFT QUERY-ID
=
NO
YES
COMPARE CFT-QUESTION NO. TO QCF-ENTRY QUESTION NO.
EQUAL
YES
8A
NO
>
NO
8C
YES
7


8A
EXAMINE QCF-ENTRY GROUP TYPE
= L-Z,9, A,B,H
YES
MOVE QCF-ENTRY TO OUTREC
WRITE OUTREC
7B
NO
= J,C,D
NO
SET UP ERROR MSG
CLOSE FILE CONTENT DIRECTORY
8B
CLOSE OUTPUT FILE
RETURN
YES
SEARCH L-TABLE SUBSET FOR MATCHING GROUP AND LIRC
MATCH FOUND
NO
7B
YES
MOVE QCF-ENTRY TO OUTREC
WRITE OUTREC
7B
8C
EXAMINE NEXT CFT-ENTRY
EOT
YES
NO
NEW SITE
YES
SAVE NEW SITE-ID
ADD NEW SITE-ID TO OUTREC
SAVE NEW FILE-ID
ADD NEW FILE-ID TO OUTREC
MOVE FILE TYPE AND "*" TO OUTREC
WRITE OUTREC
NEW FILE
YES
NO
COMPARE QCF-ENTRY QUESTION NO. TO CFT-ENTRY QUESTION NO.
EQUAL
YES
NO
CFT ENTRY >
YES
MOVE 0 TO QCF-TABLE INDEX
NO
7B
8

A.5 TITLE: SYNONYM RESOLUTION MODULE (SRM)

A.5.1 PURPOSE

By means of the Synonym Resolution File (SRF), SRM translates input constraint LIRCs and values of a query to the equivalent LIRCs and values which are contained in the data base(s) being processed. Since each site has its own SRF, ECM will call SRM once for each site.

A.5.2 INPUTS

1. IN-CONSTRAINTS - Query Constraint File (QCF) as it was reformatted and output by the File Selection Module (FSM). File contains 80-character fixed length records.

2. SYNONYM-FILE - Synonym Resolution File (SRF) for the processing site. File contains 800-character fixed length records. Currently, there is no special module to create or update a SRF for any site. SRF's to date have been created by reading in 10 data cards and creating an 800-character ISAM record from them.

A.5.3 OUTPUT

CONSTRAINTS-OUT - QCF as revised by SRM. Records are still 80-character fixed length. Appropriate LIRCs and data values have been replaced with synonymous LIRCs and values, where appropriate.

A.5.4 RECORD FORMATS - (See Figures A1-A5 and A-10)

Section & Page 101.
Subsection
Date Documented
Change Notice #

Fold back at dotted line.

CARD OR RECORD LAYOUT - DOUBLE

SYSTEM ID: ☐ New ☐ Revision

RECORD TITLE:
FILE ID: SYNONYM RESOLUTION FILE
RECORD LENGTH: 14 MINIMUM ☑ FIXED ☐ 800 MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR 1 RECORD/BLOCK

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	
DELETE CODE	TOTAL CHARS	SYNONYM KEY: FILE ID	SYNONYM KEY: I/P LIRC	SYNONYM KEY: TIE	CONTIN	FIRST REC: LIRC	FIRST REC: INV	FIRST REC: TYPE	FIRST REC: DEC	FIRST REC: COUNT	SEE REDEFINITION OF "FIRST REC" BELOW

(1) SPACE OR HIGH VALUE
(2) '014' ... '800'
(3) ALPHANUMERIC
(4) NUMERIC
(5) ALPHANUMERIC
(6) SPACE OR 'X'
(7) NUMERIC
(8) '00' ... '99'
(9) 'I' 'R' 'N' 'A' 'L' OR SPACE
(10) NUMERIC OR SPACE
(11) NUMERIC

RECORD TITLE: REDEFINITION OF "FIRST REC"
FILE ID: SYNONYM RESOLUTION FILE
RECORD LENGTH: 12 MINIMUM ☑ FIXED ☐ 784 MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

	(1)	(2)	(3)	(4)	(5)			
	SYN CNT	SYNONYM VALUE	NO INP VAL	INP CNT 1	INPUT VALUE 1	INP CNT 2	INPUT VALUE 2	. . .

(1) NUMERIC
(2) VARIABLE LENGTH
(3) NUMERIC
(4) NUMERIC
(5) VARIABLE LENGTH

Figure A-10

A.5.4.1 SYNONYM RESOLUTION FILE (SRF)

A. Fixed Part

	Data Field	Position	Picture	Values
1.	Delete Code	1	X	SPACE or H-V
2.	Total Chars	2-4	9(3)	014-800
3.	Synonym Key			
	a. File ID	5-7	X(3)	ALPHAN.
	b. I/P LIRC	8-11	X(4)	NUMERIC
	c. Tie Breaker	12-13	X(2)	ALPHAN. (L-V for 1st Rec of Key)
4.	Continuation	14	X	SPACE or "X"

- For continuation records skip rest of fixed part and go to variable part -

	Data Field	Position	Picture	Values
5.	Synonym LIRC	15-18	X(4)	NUMERIC
6.	Inversion Ind	19-20	9(2)	00-09
7.	Type	21	X	"I","R","N","A", "L" or SPACE
8.	No. Decimals	22	X	NUMERIC or SPACE
9.	Count	23-26	9(4)	NUMERIC

NOTE:

1. Continuation = Space (Last or only Rec for Key)
   = "X" (At least one more Rec)

2. Inversion Ind = 00 (Syn. LIRC is not inverted)
   = 01-09 (Number of inverted Chars for Syn LIRC)

3. Type = I Syn. LIRC is Integer
   = R Syn. LIRC is Real
   = N Syn. LIRC is Numeric
   = A Syn. LIRC is Alphabetic
   = L Syn. LIRC is Alphanumer.

4. No. Decimals = 00-09 (No. of Decimal places in Real Synonym Value)

= Space - Syn. Value not Real

5. Count = Max. size for Syn. Value (No. of characters)

B. Variable Part

1. Char. Count Synonym Value - 01-99
2. Synonym Value
3. No. of Input Values - 01-99
4. Char. Count I/P Value - 1 - 01-99
5. I/P Value - 1
6. Char. Count I/P Value - 2 - 01-99
7. I/P Value - 2

.
.
.

Char. Count I/P Value - N - 01-99
I/P Value - N

- Continue with B-1 above -

NOTE: Variable part of continuation records must start with:
1) Char. Count Synonym Value
or 2) No. of Input Values
or 3) Char. Count Input Value

A.5.5 *STORAGE REQUIREMENTS*

Program (less library modules) occupies 10,000 bytes on UNIVAC Series 70/45.

A.5.6 PROCESSING DESCRIPTION (Refer to SRM Flowchart, page 87)

A.5.6.1 GENERAL DESCRIPTION

The Synonym Resolution Module (SRM) begins processing by reading and writing Query Constraint File (QCF) records until the site record which matches the site value passed by the Executive Control Module (ECM) is found. If a matching site record is not found, SRM sets an error indicator and passes an error message to ECM.

All C, D, J, and L-Z records are processed for all files of the site as they are encountered. After the entire site has been processed, the remaining old QCF records (if any) are written unchanged to the new QCF.

A.5.6.2 ACCESSING THE SYNONYM RESOLUTION FILE

To access the Synonym Resolution File (SRF), SRM sets up a key consisting of file identification, QCF LIRC, and a tie-breaker field. If the key for a given LIRC of an old QCF record is not found on the SRF, the old QCF record is written on the new QCF unchanged and the next QCF record is read.

A.5.6.3 PROCESSING C RECORDS

A C record may have one or two LIRCs which are checked by SRM. When a match is found for a C LIRC on the SRF, the synonymous LIRC, type, and number of decimals fields will replace the original ones on the QCF record.

A.5.6.4 PROCESSING D RECORDS

For a D record with a matching LIRC on the SRF, the synonymous LIRC and count field replace the original ones on the QCF record.

A.5.6.5 PROCESSING J RECORDS

For a J record with a matching LIRC on the SRF, the synonymous LIRC replaces the original one on the QCF record.

A.5.6.6 PROCESSING L-Z RECORDS

a. General Description - For a record with a type of L through Z and a matching LIRC on the SRF, the synonymous LIRC and inversion indicator replace the original ones on the QCF record. Then the remainder of the SRF record is checked for a match between the QCF search value and an SRF input value.

(Note: SRF input value refers to a specific data field on the SRF record.)

b. Description of QCF and SRF Values - A QCF record will have only one search value unless it is a ranging search (BE, BN, BL, or BH) in which case it will have exactly two search values. However, more than one SRF input value may correspond to the same synonym value and there may be any number of synonym values on the SRF for a given LIRC.

On the SRF record, each input value and synonym value field is immediately preceded by a character count (CC) field which indicates how long a given value is. (These fields are all variable in length.) There is also a field containing the number of input values for a synonym value. This field precedes the CC field of the first input value for the synonym value. A total number of characters (TOT-CHARS) field, which contains the size of the entire SRF record, is in the fixed portion of the record. For the first record of a key, the fixed part is the first 26 characters. For all continuation records, it is the first 14.

c. Search for Matching Values - Since a QCF record may have two search values which will be checked at the same time, two hit indicators (HIT1 and HIT2) are used. HIT2 will be set to 1 (signifying a match has already been found) if no second search value occurs on the QCF record. Before beginning to search for matching values, the character index (INDEXS) will be set to 26 (the size of the fixed part of the first record of the key.)

If no values are on the SRF record (TOT-CHARS = 26), then the original search value is retained and the search is terminated.

If, however, synonym and input values are present on the SRF record, then the following steps occur:

(1) CC field for synonym value is retrieved

(2) Synonym value is stored

(3) Number of input values field is stored

(4) CC field for input value is retrieved

(5) Input value is stored

(6) If HIT1 = 0, the input value (from Step 5) is compared to the first search value of the QCF record. If a match occurs, the synonym value (from Step 2) replaces the first search value on the new QCF record and HIT1 is set equal to 1.

(7) If HIT2 = 0, Step 6 is carried out for the second search value and HIT2 is set equal to 1.

After step 7 has been executed, the process continues with Step 4, retrieving subsequent input values and testing them, until both HIT1 and HIT2 = 1 or no more input values exist for the synonym LIRC. If the former occurs, the searching is terminated. In the latter case, the process continues by going back to Step 1 and retrieving the next synonym value. The search for a match then proceeds as it did for the first synonym value. If, however, there are no more synonym values for the key, then a warning is issued (via ECM) and the original search value is retained on the new QCF record.

At Steps 1, 3, and 4, a check is made to determine whether the end of the SRF record has been reached (i.e. INDEXS = TOT-CHARS). When this occurs, the continuation code is checked and the next record is read, if appropriate.

(Note: A continuation record may not begin with the synonym value or input value fields. It must begin with the CC field for the synonym value or the CC field for the input value or the number of input values field.)

When the search for matching values terminates, processing continues by reading the next QCF record. (See A.4.7.1) GENERAL DESCRIPTION).

A.5.6.7 UNIVAC VS IBM

The only difference between the UNIVAC and IBM versions of SRM is in accessing the SRF. The UNIVAC uses a READ...INVALID KEY statement, while the IBM uses a START statement followed by a READ... AT END. The KEY of the UNIVAC version is the one described in G.2 ACCESSING THE SYNONYM RESOLUTION FILE. For the IBM START, the KEY is only the file identification and the QCF LIRC. The tie breaker field is not used.

To date, the IBM version has been tested only as a stand-alone process.

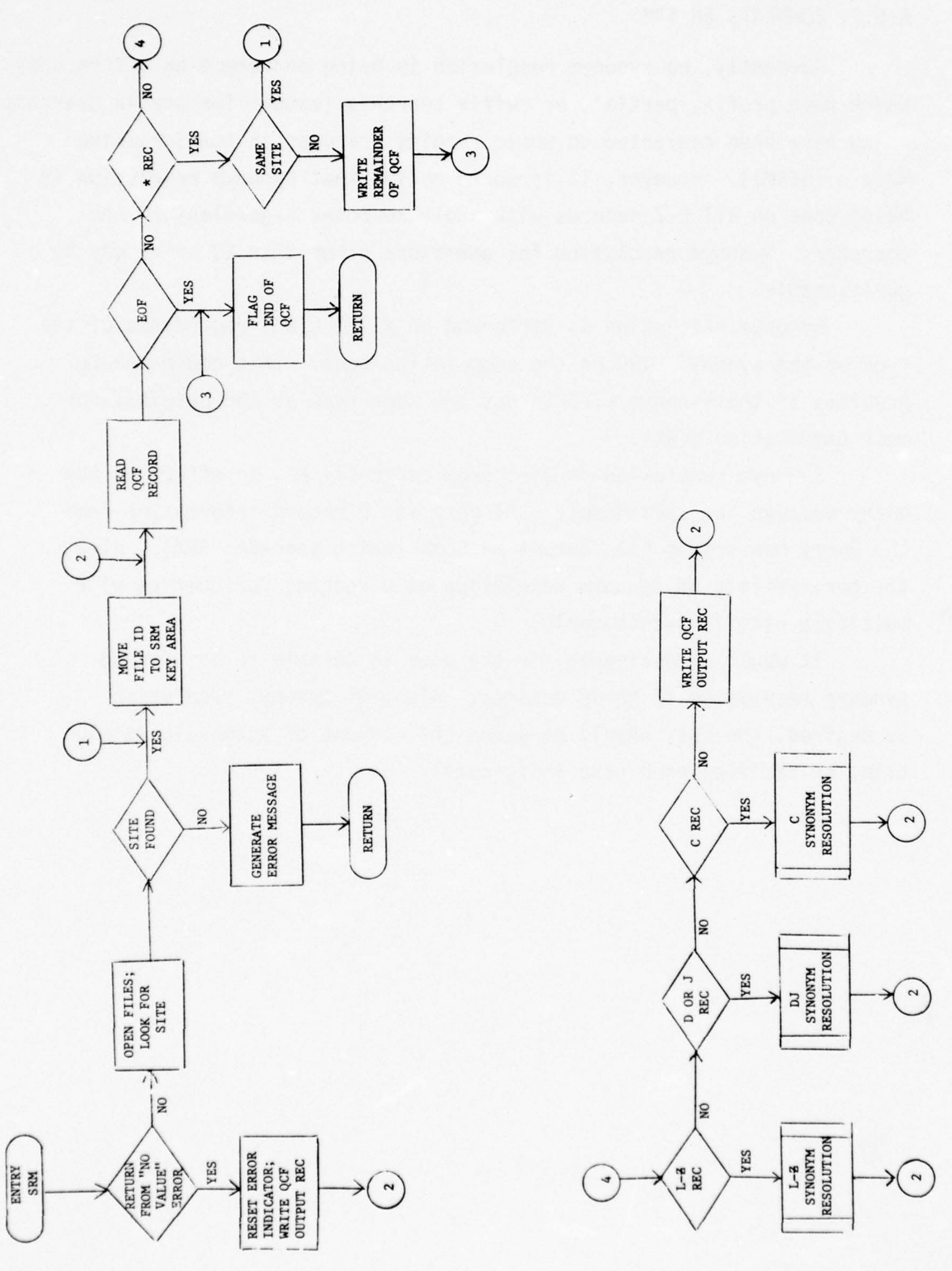
A.5.7 COMMENTS ON SRM

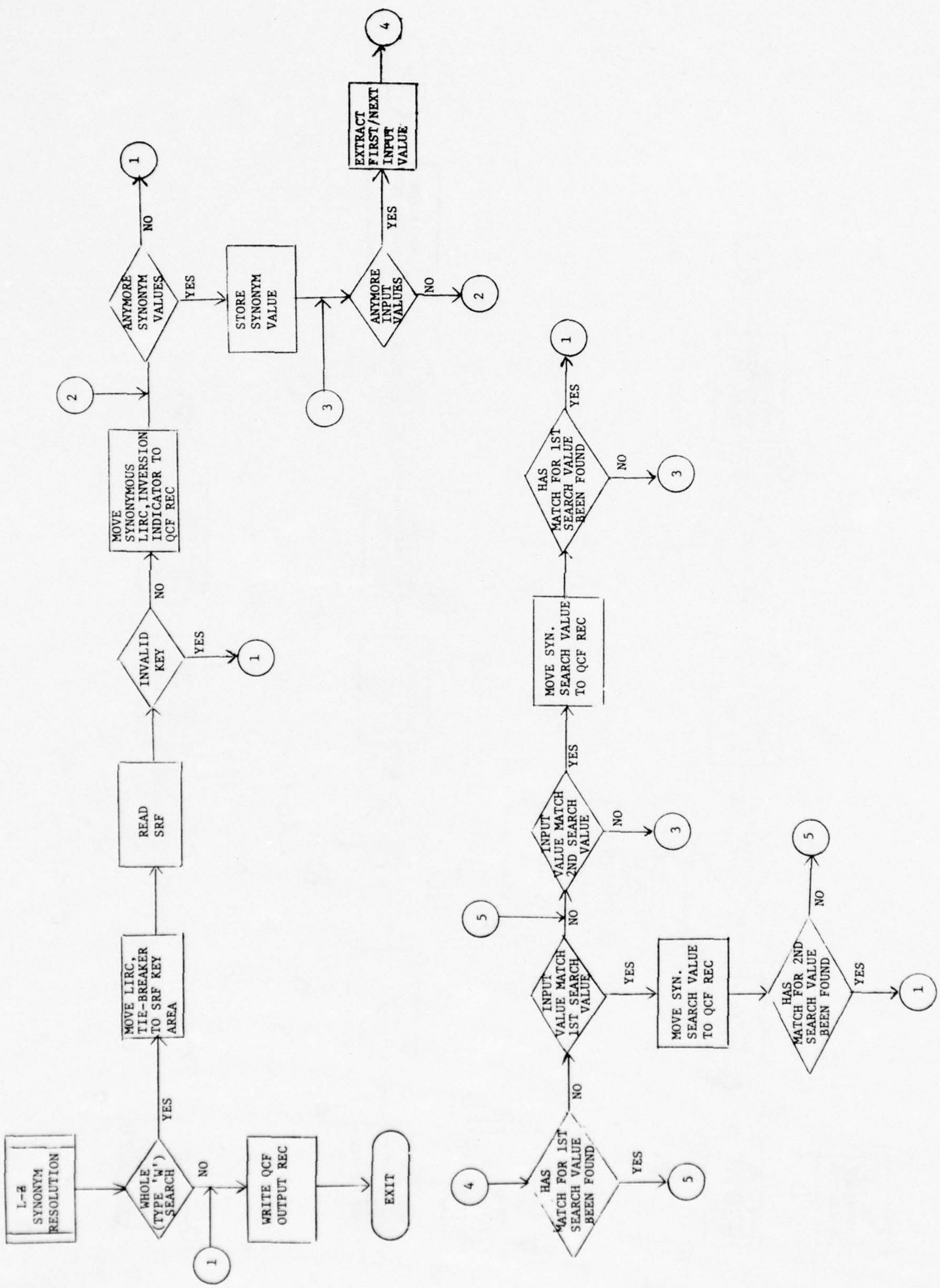
Currently, no synonym resolution is being performed on L-Z records which have prefix, partial, or suffix searches (except for prefix searches which have been converted to whole ranging searches in the Screening Module (SCRM)). However, it is worth noting that synonym resolution is being done on all L-Z records with whole searches regardless of the operator. Synonym resolution for operators other than EQ or NE may be questionable.

Synonym resolution is performed on all C LIRCs regardless of the type of the synonym LIRC or the computation code. This could create problems if the synonym LIRC is not the same type as the original for most computation codes.

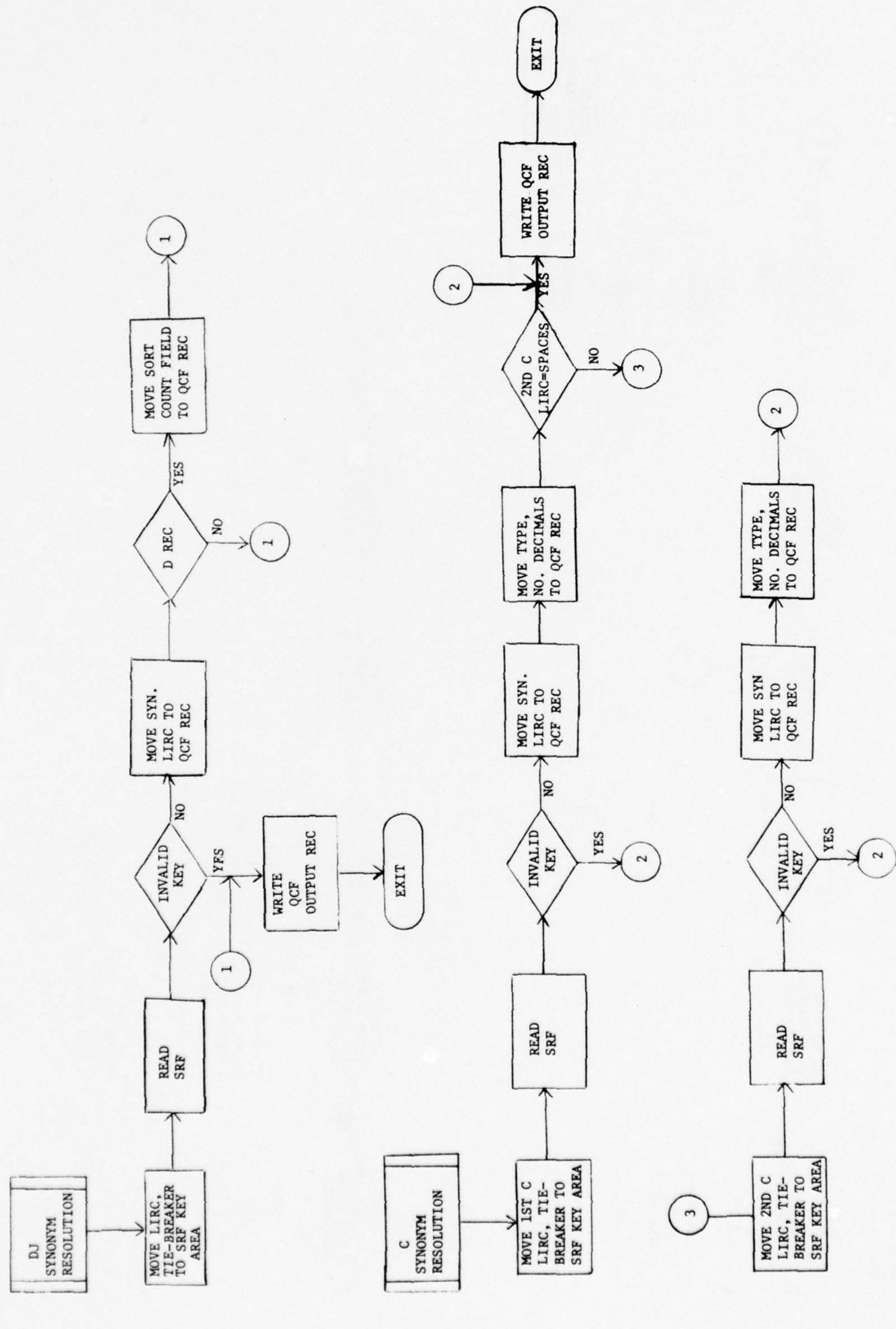
Synonym resolution on D records currently has no effect on the query because the Sort Module (SM) gets its D record information from the Query Constraint File output by SCRM (which precedes SRM). Also, the desirability of synonym resolution on D records for queries with multifile hits is questionable.

It would be desirable for the user to be able to opt for no synonym resolution if he so desires. Also, if synonym resolution is desired, the user should be given the options of automatic resolution, or deciding each case individually.


ENTRY SRM
RETURN FROM "NO VALUE" ERROR
YES
NO
RESET ERROR INDICATOR; WRITE QCF OUTPUT REC
2
OPEN FILES; LOOK FOR SITE
SITE FOUND
NO
GENERATE ERROR MESSAGE
RETURN
YES
1
MOVE FILE ID TO SRM KEY AREA
2
READ QCF RECORD
EOF
YES
3
FLAG END OF QCF
RETURN
NO
* REC
NO
4
YES
SAME SITE
YES
1
NO
WRITE REMAINDER OF QCF
3
4
L-Z REC
YES
L-Z SYNONYM RESOLUTION
2
NO
D OR J REC
YES
DJ SYNONYM RESOLUTION
2
NO
C REC
YES
C SYNONYM RESOLUTION
2
NO
WRITE QCF OUTPUT REC
2



L-2 SYNONYM RESOLUTION
WHOLE (TYPE 'W') SEARCH
YES
NO
MOVE LIRC, TIE-BREAKER TO SRF KEY AREA
READ SRF
INVALID KEY
YES
NO
MOVE SYNONYMOUS LIRC, INVERSION INDICATOR TO QCF REC
ANYMORE SYNONYM VALUES
NO
YES
STORE SYNONYM VALUE
ANYMORE INPUT VALUES
YES
NO
EXTRACT FIRST/NEXT INPUT VALUE
WRITE QCF OUTPUT REC
EXIT
HAS MATCH FOR 1ST SEARCH VALUE BEEN FOUND
YES
NO
INPUT VALUE MATCH 1ST SEARCH VALUE
YES
NO
MOVE SYN. SEARCH VALUE TO QCF REC
HAS MATCH FOR 2ND SEARCH VALUE BEEN FOUND
YES
NO
INPUT VALUE MATCH 2ND SEARCH VALUE
NO
YES
MOVE SYN. SEARCH VALUE TO QCF REC
HAS MATCH FOR 1ST SEARCH VALUE BEEN FOUND
NO
YES
1
2
3
4
5


DJ SYNONYM RESOLUTION
MOVE LIRC, TIE-BREAKER TO SRF KEY AREA
READ SRF
INVALID KEY
YFS
NO
1
WRITE QCF OUTPUT REC
EXIT
MOVE SYN. LIRC TO QCF REC
D REC
NO
1
YES
MOVE SORT COUNT FIELD TO QCF REC
1
C SYNONYM RESOLUTION
MOVE 1ST C LIRC, TIE-BREAKER TO SRF KEY AREA
READ SRF
INVALID KEY
YES
2
NO
MOVE SYN. LIRC TO QCF REC
MOVE TYPE, NO. DECIMALS TO QCF REC
2ND C LIRC=SPACES
NO
3
YES
2
WRITE QCF OUTPUT REC
EXIT
3
MOVE 2ND C LIRC, TIE-BREAKER TO SRF KEY AREA
READ SRF
INVALID KEY
YES
2
NO
MOVE SYN LIRC TO QCF REC
MOVE TYPE, NO. DECIMALS TO QCF REC
2

A.6 TITLE: PRELIMINARY HIT RESOLUTION MODULE (PHRM)

A.6.1 PURPOSE

The Preliminary Hit Resolution Modules (PHRM) will first determine whether preliminary or complete hit resolution can be done for the given data base for all the appropriate questions of the query. If so, then PHRM creates a file of hits and/or hit candidates. If not, the module will be bypassed and all hit resolution for that data base will be done by the Secondary Hit Resolution Module (SHRM).

A.6.2 INPUTS

A.6.2.1 CONSTRAINTS

Query Constraint File (QCF) as it was generated by the Synonym Resolution Module (SRM). File contains 80-character fixed length records.

A.6.2.2 INV-MF(2)

Inverted Master File (INV-MF) created by the Invert Update Program. File contains 644-character fixed length records in the UNIVAC version and 293-character fixed length records with 10 records per block in the IBM version.

A.6.2.3 REC-DESCRIPTION-FILE

The Record Description File (RDF) is created by the Invert Update Program. File contains 46-character fixed length records with 50 records per block.

A.6.3 INTERMEDIATE FILES

A.6.3.1 FILE-A, FILE-B, FILE-C, FILE-D, FILE-E, FILE-F

These are scratch files used in the hit merging procedures. All contain 54-character fixed length records. The IBM version blocks

the records at 10 per block; UNIVAC version uses the default blocking.

A.6.3.2 QSRT-1, QSRT-2

These are scratch files used in the "quick" sort/merge procedure. Both contain 12-character fixed length records with 64 records per block. Currently, these two files exist only in the IBM version. They are part of Program Change Proposal (PCP) #NC0118 which has not yet been put into the UNIVAC version.

A.6.4 OUTPUT

FILE-A, FILE-B, FILE-C, FILE-D, FILE-E, or FILE-F - The output of PHRM will be one of the above six files depending on where the final merged data ended. If, however, the data base being queried is a random file, the output file will be FILE-F. The above files are the same files as the intermediate files which have the same names.

Currently, the NAVLIS system can handle only FILE-F as the output file.

A.6.5 RECORD FORMATS - See Figures A1-A5, A11-A13

A.6.6 STORAGE REQUIREMENTS

Program (less library modules) occupies 42,000 bytes on UNIVAC Series 70/45.

A.6.7 PROCESSING DESCRIPTION (Refer to PHRM Flowchart, page 108)

In the following description of PHRM, parenthesis ( ) are used to enclose statements which refer to the IBM version only.

A.6.7.1 GENERAL DESCRIPTION

PHRM begins by determining whether any preliminary hit resolution can be done for the data base being processed. If no resolution can be done, then control is returned to the Executive Control Module (ECM) with an indicator set to signify no hit resolution has been done. There is no further processing by PHRM for that data base.

Section & Page 101.

Subsection

Date Documented

Change Notice #

← Fold back at dotted line.

CARD OR RECORD LAYOUT – DOUBLE

SYSTEM ID: ☐ New ☐ Revision

RECORD TITLE:

FILE ID: INVERTED MASTER FILE

RECORD LENGTH: 32 MINIMUM ☑ FIXED ☐ 444 MAX.

WORDS ☐ BYTES ☐ CHARACTERS ☑

BLOCKING FACTOR 1 RECORD / BLOCK

MEDIUM:

CARD ☐ TAPE ☐ DASD ☐

OTHER ☐

DENSITY

TRACKS: 7 ☐ 9 ☐

CARD STOCK

CHARACTER ENCODING:

HOL ☐ EXH ☐

BCD ☐ EBCDIC ☐ ASCII ☐

PACKED DECIMAL ☐

BINARY OR NON-CHARACTER INFORMATION ☐

OTHER ☐

PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:

TRAILER:

COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

DELETE CHAR	START LRN	NO LRN	SEARCH KEY: LIRC	SEARCH KEY: DATA VALUE	SEARCH KEY: KEY ID	LINE CNT	CONT CODE
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

INV-DATA
NINE CHARACTER DATA FIELDS OCCURRING 1 TO 30 TIMES (IBM)
OR 1 TO 69 TIMES (UNIVAC) DEPENDING ON NO-LRN
DATA-VALUE
0 TO 8 INV-DATA FIELDS (1 LESS THAN START-LRN)
LIST OF LRN'S
1 TO 30 (IBM) OR 1 TO 69 (UNIVAC)

(1) SPACE OR HIGH VALUE
(2) NUMERIC
(3) NUMERIC
(4) NUMERIC
(5) ALPHANUMERIC
(6) ALPHANUMERIC
(7) NUMERIC OR SPACES
(8) SPACE OR 'X'

RECORD TITLE:

FILE ID:

RECORD LENGTH: MINIMUM ☐ FIXED ☐ MAX.

WORDS ☐ BYTES ☐ CHARACTERS ☐

BLOCKING FACTOR

MEDIUM:

CARD ☐ TAPE ☐ DASD ☐

OTHER ☐

DENSITY

TRACKS: 7 ☐ 9 ☐

CARD STOCK

CHARACTER ENCODING:

HOL ☐ EXH ☐

BCD ☐ EBCDIC ☐ ASCII ☐

PACKED DECIMAL ☐

BINARY OR NON-CHARACTER INFORMATION ☐

OTHER ☐

PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:

TRAILER:

COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Figure A-11

Fold back at dotted line.

Section & Page 101.
Subsection
Date Documented
Change Notice #

CARD OR RECORD LAYOUT – DOUBLE

SYSTEM ID: New   Revision

RECORD TITLE:
FILE ID: RECORD DESCRIPTION FILE
RECORD LENGTH: 46   MINIMUM ☐   FIXED ☑   MAX.
WORDS ☐   BYTES ☐   CHARACTERS ☑
BLOCKING FACTOR 1 RECORD / BLOCK

MEDIUM:
CARD ☐   TAPE ☐   DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐   9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐   EXH ☐
BCD ☐   EBCDIC ☐   ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON–CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7–TRACK BCD) ☐   ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Positions	1	2–10	11–40	41–46
Field	DELETE	LOGICAL RECORD NUMBER	PHYSICAL RECORD KEY	PHYS. REC/LOG. REC
	(1)	(2)	(3)	(4)

(1) SPACE OR HIGH VALUE
(2) ALPHANUMERIC
(3) ALPHANUMERIC
(4) SPACES OR NUMERIC

RECORD TITLE: INTERMEDIATE SCRATCH FILES : A-F
FILE ID: HIT RECORD FILE
RECORD LENGTH: 54   MINIMUM ☐   FIXED ☑   MAX.
WORDS ☐   BYTES ☐   CHARACTERS ☑
BLOCKING FACTOR 1 RECORD / BLOCK

MEDIUM:
CARD ☐   TAPE ☐   DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐   9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐   EXH ☐
BCD ☐   EBCDIC ☐   ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON– CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7–TRACK BCD) ☐   ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Positions	1–3	4–6	7–15	16–18	19–48	49–54
Field	QUERY ID	QUEST NO	LOGICAL RECORD NUMBER	LINE CNT	PHYSICAL RECORD KEY	PHYS. REC/LOG. REC
	(1)	(2)	(3)	(4)	(5)	(6)

(1) ALPHANUMERIC
(2) NUMERIC
(3) ALPHANUMERIC
(4) SPACES OR NUMERIC
(5) ALPHANUMERIC
(6) SPACES OR NUMERIC

Figure A-12

Section & Page 101.
Subsection
Date Documented
Change Notice #

Fold back at dotted line.

CARD OR RECORD LAYOUT – DOUBLE

SYSTEM ID:

☐ New ☐ Revision

RECORD TITLE: HIT-REC
FILE ID: HIT FILE

RECORD LENGTH: 78 MINIMUM ☐ FIXED ☑ MAX.

WORDS ☐ BYTES ☐ CHARACTERS ☑

BLOCKING FACTOR 10 RECORDS/BLOCK

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Positions	Field
1–3	QUERY ID (1)
4–6	QUEST NO (2)
7–8	SCH TYPE
9	CONSTR (3)
10–11	SET NO (4)
12–14	FILLER
15	W/P SWITCH
16–19	LIRC (5)
20–26	FILLER
27	PARALLEL (6)
28	EFFICIENCY (7)
29–78	SEARCH VALUE

(1) ALPHANUMERIC
(2) NUMERIC
(3) "L" ... "Z"
(4) NUMERIC
(5) NUMERIC
(6) "P" OR SPACE
(7) 1 OR SPACE

RECORD TITLE:
FILE ID:

RECORD LENGTH: MINIMUM ☐ FIXED ☐ MAX.

WORDS ☐ BYTES ☐ CHARACTERS ☐

BLOCKING FACTOR

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Figure A-13

If, however, preliminary hit resolution can be done, then PHRM begins the hit resolution by opening files and setting the necessary indicators and tables. Then, for each question, the question resolution indicator is tested. If it is an "N" (no resolution), then that question is bypassed and the next one processed. If it is a "P" (partial) or "C" (complete), then the hits from the previously processed question are added to the final hit file, unless that question is to be merged with the current one. In that case nothing is added to the final hit file until the entire merged question is processed. Next, the FIND-HITS procedure is performed for the current question. After each question is processed, PHRM will check to see if there are any more questions for the data base. If there are no more, then the remaining hits (if any) are written to the final hit file and control returns to ECM. Otherwise, processing continues as before with the next question and goes on until none are left.

A.6.7.2 PHRM-DET

PHRM determines whether any preliminary hit resolution is possible by first finding the QCF file ID record for the data base being processed. If it is not found, then control returns to ECM. Otherwise, PHRM goes through all the constraints for the data base, storing the A and L - Z records and setting the group and question resolution indicators. A group can have either complete (C) or no (N) hit resolution. Complete group resolution occurs only when all constraints for the group are inverted (i.e., inversion indicator = 0). A question can have either complete (C), partial (P), or no (N) hit resolution. If all groups for a question are "C", then the question is "C". If all groups are "N", then the question is "N". For questions having groups with combinations of "Cs" and "NS", the resolution will be "P". Currently, no preliminary hit resolution is done on any question for the data base if at least one question has no hit resolution.

When the next file ID record is read, PHRM moves it to the NEXT-FILEID field for ECM. When this occurs or when an EOF is read on QCF, the question resolution indicator for the last question is set. If no preliminary hit resolution is to be done, control will return to ECM at this point.

A.6.7.3 FIND-HITS

a. Preliminary - For each question to be processed, the FIND-HITS Section first initializes indicators. Then the group resolution indicator is tested for the first group. If it is an "N", that group is by-passed and the group resolution indicator of the next group is checked. When a group with a "C" resolution is found, the search value(s) from the first constraint of the group is (are) moved to the corresponding Working-Storage areas for testing.

b. Create INV-MF Key - Then the key used to read the INV-MF is created. For most searches, this key consists of the LIRC followed by the first nine inversion characters of the (first) search value and a two-character tie-breaker field which is initialized as low-values. If the search value is inverted on less than nine characters, then the remainder of that field is low-value filled. Also, for LE, LS, and NE operation codes (op codes) or any non-whole searches, the inversion character field is all low-values.

c. Read INV-MF - (For the IBM version, random access is achieved by a START ... INVALID KEY on the INV-MF. The key is actually a generic key. It does not contain the tie-breaker field or any of the low-value filler of the inversion field. At the very least, the key will be the LIRC. When an invalid key occurs, PHRM attempts another START using one less character from the right side of the key if the op code was not EQ or the LIRC was equal to "0000" or the key was not just the LIRC. Should an invalid key occur again, the process continues until either the key is valid, or the invalid key consists only of the LIRC. When a valid key is found, processing

continues with reading the INV-MF record. Otherwise, PHRM goes to the Next-Constraint procedure (Section A.6.7.3F)).

For the UNIVAC version, the READ ... INVALID KEY format is used. When the invalid key occurs, the next record is read sequentially. If it is for the proper LIRC, it goes to the Searches procedure (Section A.6.7.3d); otherwise, processing resumes with the Next-Constraint (Section A.6.7.3f).

d. Searches - After a valid read is executed, the full value of the INV-MF data value field is moved to a Working-Storage area. (For a suffix search, the data value is shifted to the left until the number of data characters remaining equals the number in the suffix search value. Then the right side of the data value field is space-filled.)

For a prefix or partial search, the characters of the data value which correspond to those characters of the search value that are equal to asterisks (*) are changed to asterisks.

(Then, that portion of the data value field which is greater than the size of the prefix or partial search value is space-filled.)

e. Test-Data-Value - The data value from the INV-MF (which may have been altered) is compared to the search value using the given op code. If a hit occurs, all the logical record numbers (LRN's) for the hit value are retrieved for subsequent merging, (see Section 5, (MERGE-HITS)) except if LIRC = 0000, when only the data value, which is also the LRN, that was a hit is retrieved. For whole searches with GE or GR op codes, subsequent LRN's for the same LIRC are also retrieved without further value tests. If the op code equals NE or the search type is not whole, all the remaining INV-MF records for the LIRC being processed are checked; otherwise, processing continues with the Next-Constraint Section A.6.7.3f). For non-hits with op codes equal to GE, GR, or NE, or those which are not whole searches, testing continues with the next INV-MF record. (For LIRC = 0000, next LRN is processed instead). All other non-hits go to Next-Constraint.

f. Next-Constraint* - (If there are no more constraints, PHRM goes to the In-Core Sort (Section A.6.7.3g). If the next question is not equal to the current one, or if the next group is not equal to the current group, the ANALIZE-GROUP logic is performed (Section A.6.7.4). Processing then continues with the In-Core Sort (Section A.6.7.3g). If both the next question and the next group are equal to the current question and group and the logical connector is an "or" (as determined by ANALIZE-GROUP), then the search value(s) is (are) moved to Working-Storage and PHRM control goes to Create INV-MF Key (Section A.6.7)).

g. In-Core Sort - For a pair of constraints not connected by an "or" operator, an in-core sort is performed on the table of hit candidates retrieved so far.

h. Merging - Then, if there are no more constraints, or if the next constraint has a different Query ID or Question Number, all levels of merging are performed (i.e. MRG - CONSTRAINTS, MRG-SETS, MRG-GROUPS, MRG-QUESTIONS). If there are no hits for the question, the no hits error message is generated and control returns to ECM. If there are hits, then processing continues with the next question. (See Section A.6.7.1 GENERAL DESCRIPTION).

If the next group is not equal to the previous group, merging is done to the MRG-GROUPS level. If there are no hits for the group (which signifies no hits for the question), then the no hits error message is generated and control returns to ECM. Otherwise, PHRM control goes to test the next group resolution indicator (Section A.6.7.3a - Preliminary).

If the next set is not equal to the previous set, then merging is done to the MRG-SETS level and processing continues with the next group resolution indicator test; otherwise only the MRG-CONSTRAINTS procedure is performed before continuing with the group resolution indicator check.

---

* For UNIVAC version, Next-Constraint begins with Merging (Section 3h).

i. "A" Record Processing - Whenever an A record is encountered, the merge indicator is set to 1 to signify a merge is in progress, or to 0 to indicate the end of a merge. After setting the indicator, the next-constraint is processed.

A.6.7.4 ANALIZE-GROUP

Whenever the next constraint to be processed is not of the same question or group as the previous constraint, then the ANALIZE-GROUP Section is performed. This section of coding checks to see that no constraints in the up-coming group are efficiency searches and all are connected by the "or" operator (i.e., set numbers are not equal). If this is the case, the group condition indicator is 0; otherwise, it is set to 1.

A.6.7.5 MERGE-HITS

When the comparison of a data value and search value(s) results in a hit, then the Merge Hits processing takes place. Each LRN from the hit INV-MF record is retrieved and, with the line count (LC), is used to create a Record ID. (The LRN is then checked to see if it is within the range of the HI/LO indicators. If it is out of range and the LIRC is equal to "0000", then some indicators are set and MERGE-HITS is completed for the LRN. if it is out of range and the LIRC is not equal to "0000", then the remaining LRN's for that record and any subsequent continuation records are checked.
If the LRN is within range, then the output hit file (OUT) is opened and the appropriate switches and indicators are initialized (if this has not already been done). Next, the size of the in-core sort table is checked. If it is already full (currently, 4000 entries), then it is sorted (see Section A.6.7.3g, In-Core Sort) and its index is reset. Then, the parallel indicator is checked. If the constraint is not parallel, the LC field is spaced out. Now, the record is moved to the next entry of the in-core sort table.)** If the LIRC is equal to

** For the UNIVAC version, an OR-MERGE is performed on the hits being retrieved for the constraint and any others which may have already been retrieved.

"0000", then MERGE-HITS is complete since PHRM must test each LRN for a LIRC "0000" constraint. Otherwise, the next LRN is retrieved and processed as before. When all the LRN's have been retrieved for a record, the continuation code is checked. If there are one or more continuation records, then the LRN's from them are also retrieved and processed. When no more LRN's are left on the last continuation record, then some indicators are set and MERGE-HITS is complete for the data value processed.

A.6.7.6 MRG-CONSTRAINTS

This procedure merges at most two files which contain sorted hits at the constraint level. That is, each file contains only the hits from one constraint record and the merging being done is for the same group and the same set. The older of the two files is denoted by CON-INDIC and the newer by HIT-INDIC (sorted output of Merge-Hits procedure). If there is no older file, then HIT-INDIC is moved to OUT (the indicator for the resultant hit file) and no actual merging is done. If, however, two files are present and neither is an efficiency file, an AND-MERGE is done. An efficiency is one which contains LRN's that are not hits. If both files are efficiency files, then an OR-MERGE is performed and the resultant file is still an efficiency file. If only one file is an efficiency file, then a NOT-MERGE is performed. For the NOT-MERGE, the resultant file is not an efficiency file. After the merging (if any), OUT becomes the new CON-INDIC. Then OUT and HIT-INDIC are reset to zero.

A.6.7.7 MRG-SETS

This procedure merges at most two files which contain sorted hits at the set level. That is, each file contains only hits from one set and the merging being done is for the same group but different sets. The older of the two files is denoted by SET-INDIC and the newer by CON-INDIC. If there is no older file, then CON-INDIC is moved to OUT and no merging is done. If, however, two files are present and neither

is an efficiency file. an OR-MERGE is done. If both files are efficiency files, then an AND-MERGE is performed and the resultant file is an efficiency file. If only one file is an efficiency file, then a NOT-MERGE is performed and the resultant file is still an efficiency file. After the merging (if any), OUT becomes the new SET-INDIC. Then OUT and CON-INDIC are reset to zero.

A.6.7.8 MRG-GROUPS

This procedure merges at most two files which contain sorted hits at the group level. This is, each file contains only hits from one group and the merging being done is for the same question but different groups. The older of the two files is denoted by GRP-INDIC and the newer by SET-INDIC. If there is no older file, then SET-INDIC is moved to OUT and no actual merging is done. If two files are present and neither is an efficiency file, then an AND-MERGE is done. If, however, both files are efficiency, then an OR-MERGE is performed and the resultant file is an efficiency file. If only one file is an efficiency file, then a NOT-MERGE is performed and the resultant file is not an efficiency file. After the merging (if any), OUT becomes the new GRP-INDIC. Then the HI/LO indicators are set and OUT and SET-INDIC are reset to zero.

A.6.7.9 MRG-QUESTIONS

This procedure merges at most two files which contain sorted hits at the question level. That is, each file contains only hits from one question and the merging being done is for the same question or set of merged questions. The older of the two files is denoted by QST-INDIC and the newer by GRP-INDIC. If there is no older file, then GRP-INDIC is moved to OUT and no actual merging is done. If, however, two files are present, then an OR-MERGE is done. After the merging (if any), OUT becomes the new QST-INDIC. Then OUT and GRP-INDIC are reset to zero. If the question processed is not part of a merged question, if it is the last in a merged question, or if

there are no more questions to process, then PHRM goes to WRITE-HIT-FILE; otherwise, it processes the next question.

A.6.7.10 OR-MERGE

a. General Description - For the OR-MERGE (and the AND-MERGE), IN1 (the first input file) represents the older of the two hit candidate files being merged, and IN2 (the second input file) is the newer. For all merges, OUT is the resultant output file. In the OR-MERGE, duplicate LRN's with the same LC are eliminated (except if two questions with partial hit resolution are being merged). If the LRN's are the same but the LC's are different, both hits are retained unless exactly one is a parallel record (LC $\neq$ spaces), in which case only the parallel record is retained. Two exceptions are: (1) If the parallel one had only partial hit resolution but the non-parallel had complete, then both would be retained, and (2) if both input files are efficiency files, then the non-parallel one is retained.

b. Preliminary Processing - If there is no IN2 file, then all of IN1 is copied to OUT and the OR-MERGE is not performed. If IN1 and IN2 do not intersect (as determined by the HI/LO logic), then no OR-MERGE is necessary. First, all the hits from the file with the lower LRN's are written to OUT, then all the hits from the remaining file are written to OUT.

c. OR-MERGE-LOOP - PHRM merges IN1 with IN2 onto OUT in the following manner:

(1) If an end-of-file is read on either input file, then the remainder of the other file is written to OUT and the OR-MERGE is complete.

(2) If IN1-LRN < IN2-LRN, then the IN1 record is moved to OUT and the next IN1 record is retrieved.

(3) If either the IN1 file or the IN2 file are from questions which have only partial hit resolution, then go to OR-QST-MERGE (Section A.6.7.10d).

(4) If IN1-LRN > IN2-LRN, then the IN2 record will go on OUT. But, first, the LC field is spaced-filled if IN2 is not a file with parallel hits; or if two questions are being merged, the question number of the first is used. Then, the resultant IN2 record is written to OUT and the next IN2 record is retrieved.

(5) If IN1-LRN = IN2-LRN and IN1-LC = Spaces, then the IN1 record is written to OUT and the next IN1 and IN2 records are retrieved.

(6) If IN1-LRN = IN2-LRN and both IN1 and IN2 are efficiency files and either IN1-LC or IN2-LC = space (not parallel), then the next record from the file which is parallel is retrieved.

(7) If IN1-LRN = IN2-LRN and IN1-LC = spaces, then the IN2 reocrd will go on OUT. But, first, the LC field is spaced-filled if IN2 is not a file with parallel hits; or if two questions are being merged, the question number of the first is used. Then the resultant IN2 record is written to OUT and the next IN1 and IN2 records are retrieved.

(8) If IN1-LRN = IN2-LRN and IN2-LC = spaces, then the IN1 hit is written to OUT and the next IN1 and IN2 records are read.

(9) If IN1-LRN = IN2-LRN and IN1-LC < IN2-LC, then the IN1 hit is moved to OUT and the next IN1 record is read.

(10) If IN1-LRN = IN2-LRN and IN1-LC = IN2-C, then the IN1 hit is moved to OUT and the next IN1 and IN2 records are retrieved.

(11) If IN1-LRN = IN2-LRN and IN1-LC > IN2-LC, then the IN2 record will be moved to OUT. But, first if two questions are being merged, the question number of the first question is used. Then the resultant IN2 record is written to OUT and the next IN2 record read.

d. OR-QST-MERGE - (Section A.6.7.10b3 for routes which send coding here).

(1) If IN1-LRN > IN2-LRN, then the IN2 record is written to OUT and the next IN2 record is read.

(2) If IN1-LRN = IN2-LRN and IN1-LC < IN2-LC, then IN1 is written to OUT and the next IN1 record is retrieved.

(3) If IN1-LRN = IN2-LRN and IN1-LC = IN2-LC, then if both LC's are parallel and either the IN1 or IN2 file contains hits from a question with complete hit resolution, the next record of the file without complete hit resolution is read. Otherwise, both the IN1 and IN2 records are written to OUT and the next record from both input files is read.

(4) If IN1-LRN = IN2-LRN and IN1-LC > IN2-LC, then the IN2 record is written to OUT and the next IN2 record is read.

A.6.7.11 AND-MERGE

a. General Description - In the AND-MERGE, only matching hits are retained. To match, both LRN's must be the same and either both LC's are the same or one is spaces. If one is spaces, then the other LC (parallel one) is the hit.

b. Preliminary Processing - In performing an AND-MERGE, PHRM starts by first determining if there is an IN2 file. If not, then no hits are possible so the AND-MERGE is not performed.

If the IN1 and IN2 files do not intersect (as determined by the HI/LO logic), then no hits are possible and, again, no AND-MERGE is done.

c. AND-MERGE-LOOP - PHRM merges IN1 with IN2 onto OUT in the following manner:

(1) If an end-of-file is read on either input file, then the AND-MERGE is complete.

(2) If IN1-LRN IN2-LRN or IN1-LRN IN2-LRN, then nothing is put on OUT, and the next record of the file which had the lower LRN is read.

(3) If IN1-LRN = IN2-LRN and IN1-LC = IN2-LC, then the IN1 record is written to the OUT file and the next record from both input files is read.

(4) If IN1-LRN = IN2-LRN and IN1-LC ≠ IN2-LC and (IN1-LC = spaces (not a parallel file) or IN2-LC = spaces), then the parallel record is a hit. Since non-parallel records with the same LRN could exist, only the parallel input file is read again.

(5) If IN1-LRN = IN2-LRN and (IN1-LC IN2-LC or IN1-LC IN2-LC) (but neither LC = spaces), then there is no hit and either the next IN1 or IN2 record is retrieved depending on which LC was lower.

A.6.7.12 NOT-MERGE

a. General Description - For the NOT-MERGE with an AND operator (MRG-GROUPS, MRG-CONSTRAINTS), IN1 represents the input file with hit candidates and IN2 represents the efficiency input file (not hits). For the NOT-MERGE with an OR operator (MRG-SETS), the opposite is true. IN1 is the efficiency file and IN2 is the hits.

In the NOT-MERGE, the only LRN's which are candidates for the OUT file are those on IN1. IN2 is used solely to eliminate hits from IN1.

b. Preliminary Processing - If there is no IN1 file, then there are no hits so NOT-MERGE is not performed. If there is no IN2 file, or if the IN1 and IN2 files do not intersect (as determined by the HI/LO logic), then all the records on IN1 are hits and are written to OUT and the NOT-MERGE is not performed.

c. NOT-MERGE-LOOP - PHRM merges IN1 with IN2 onto OUT in the following manner:

(1) If an end-of-file is encountered on either input file, then the remainder of IN1 (if any) is written to OUT and the NOT-MERGE is complete.

(2) If IN1-LRN   IN2-LRN, then the IN1 record is written to the OUT file and the next IN1 record is read.

(3) If IN1-LRN   IN2-LRN, then nothing is put on OUT and the next IN2 record is read.

(4) If IN1-LRN = IN2-LRN and (IN-LC = IN2-LC or IN1-LC = spaces (not parallel)), then the IN1 record is not a hit, and the next IN1 and IN2 records are retrieved.

(5) If IN1-LRN = IN2-LRN and IN2-LC = spaces (not parallel), then the IN1 record is not a hit. Since the next IN1 record could also have the same LRN but with a different LC, only the next IN1 record is read.

(6) If IN1-LRN = IN2-LRN and IN1-LC   IN2-LC (but neither LC = spaces, then the IN1 record is a hit and the next IN1 record is retrieved.

(7) If IN1-LRN = IN2-LRN and IN1-LC   IN2-LC (but neither LC = spaces), then no hit exists, but the next IN2 record is read.

A.6.7.13 WRITE-HIT-FILE

This procedure writes the final hits from a question* to an output hit file. If the data base is a random file, the hits are ordered by question, so no merging is necessary. As each question is completed, its hits are written to the same file (FILE-F). If, however, the data base is sequential, then the previous output hit file (if any), which is indicated by QRY-INDIC, is merged with the present one, which is indicated by QST-INDIC. Hits are ordered by LRN, Question Number, LC. The resultant file becomes the new QRY-INDIC file.

Regardless of the data base type, a Record Description File (RDF) may be associated with the data base. If so, before the hit is transferred to the output file, the RDF file is accessed using the LRN as the key. Should an invalid key occur, an error message is generated and control returns to ECM. Eventually, it comes back to PHRM and simply by-passes the processing of the bad LRN. When a valid key is found, the physical record key (PRK) and number of physical records per logical record (NO-PR) are added to the hit record. This is done for all hits. If the data base does not have an RDF, then PRK equals LRN, and NO-PR equals spaces.

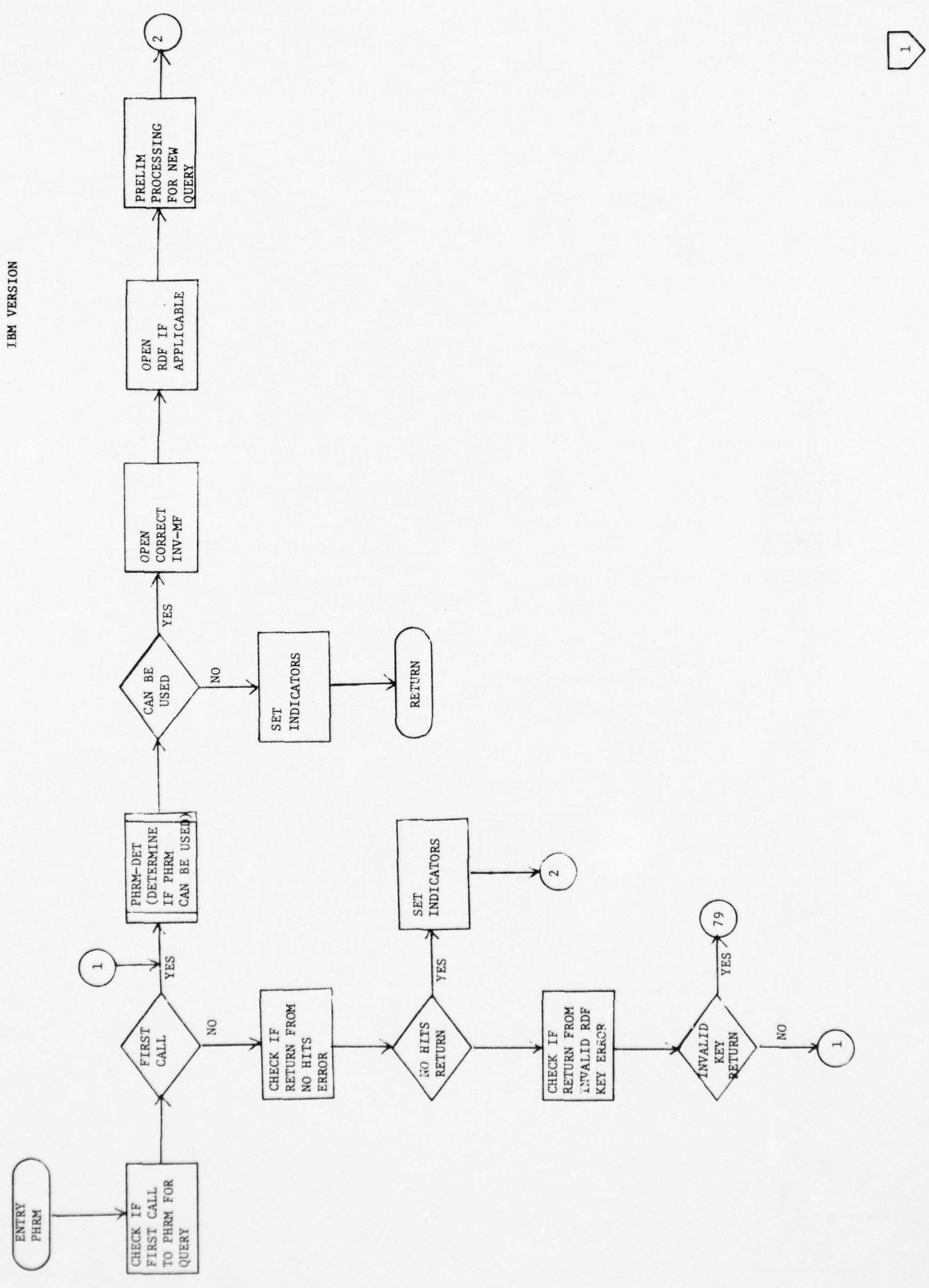
A.6.7.14 UNIVAC vs. IBM

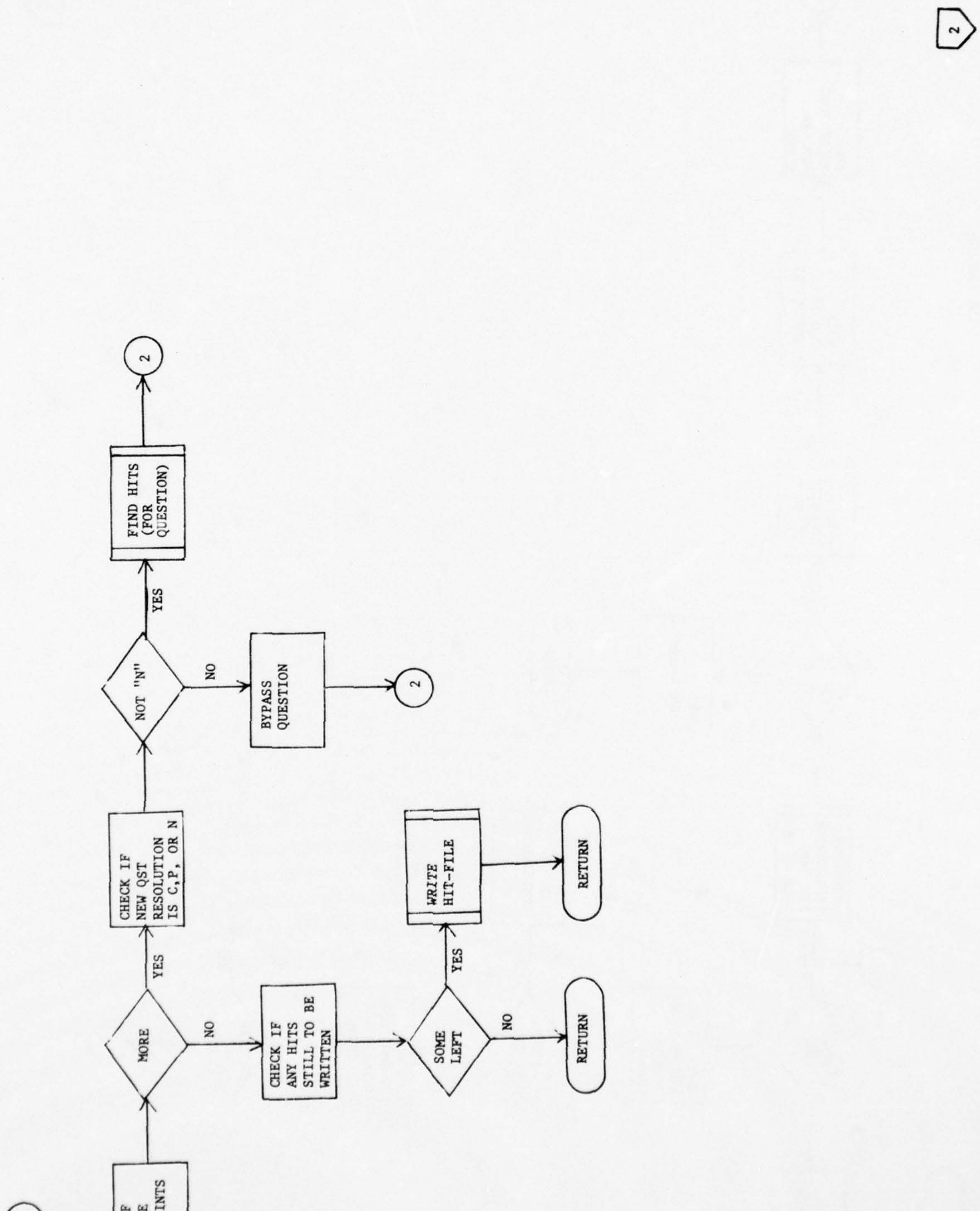
The differences between the UNIVAC and IBM versions of PHRM are as follows:

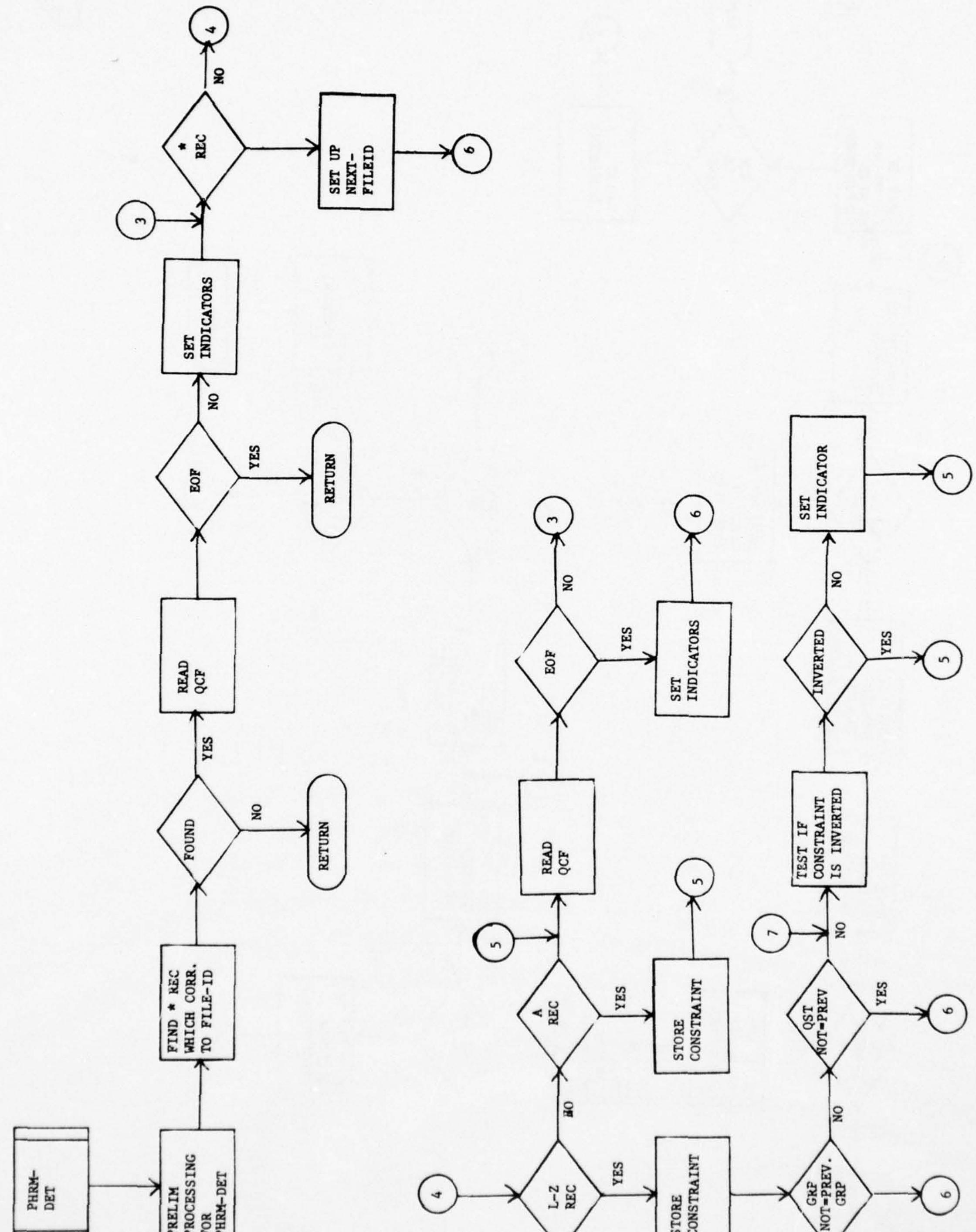
(a) The IBM version accesses the INV-MF with a START... INVALID KEY, whereas the UNIVAC version uses a READ...INVALID KEY. Also, the IBM key does not include the tie-breaker field and in some instances does not even include the entire search value. (See Read INV-MF, Section A.6.7.3c).

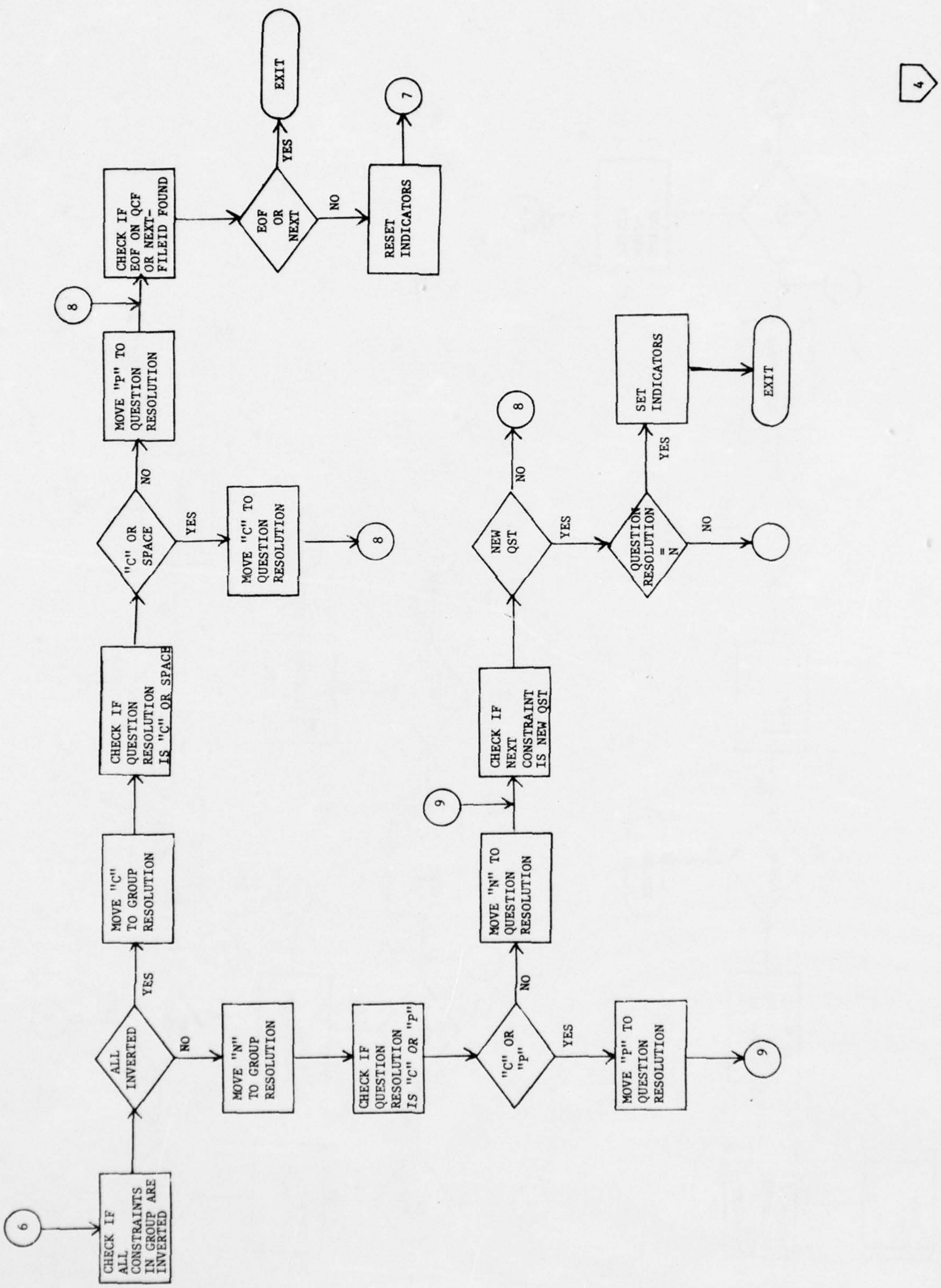
(b) The IBM version contains Program Change Proposals (PCP's) which have not yet been added to the UNIVAC version. Some PCP's correct errors and others enhance the PHRM processing.

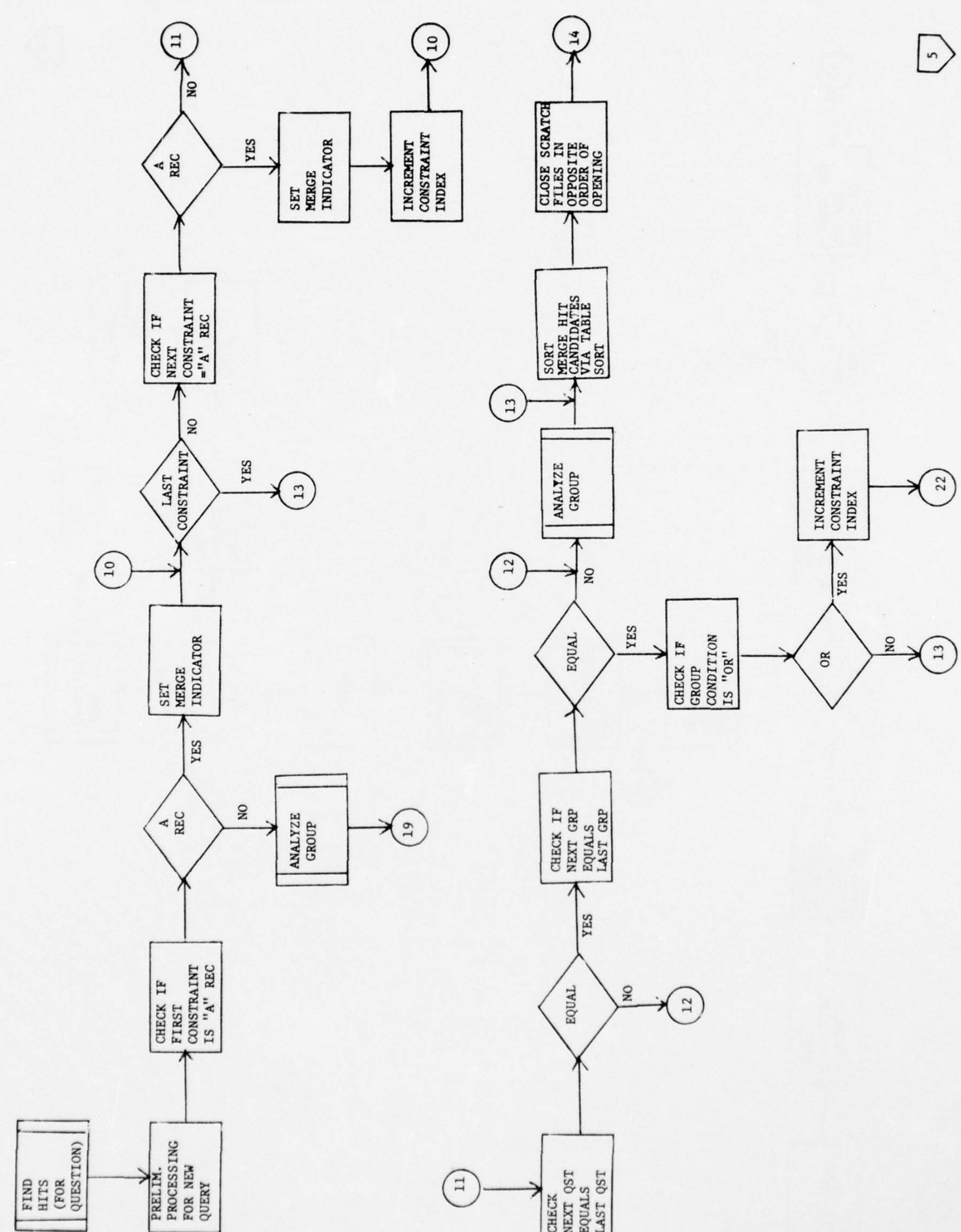
IBM VERSION

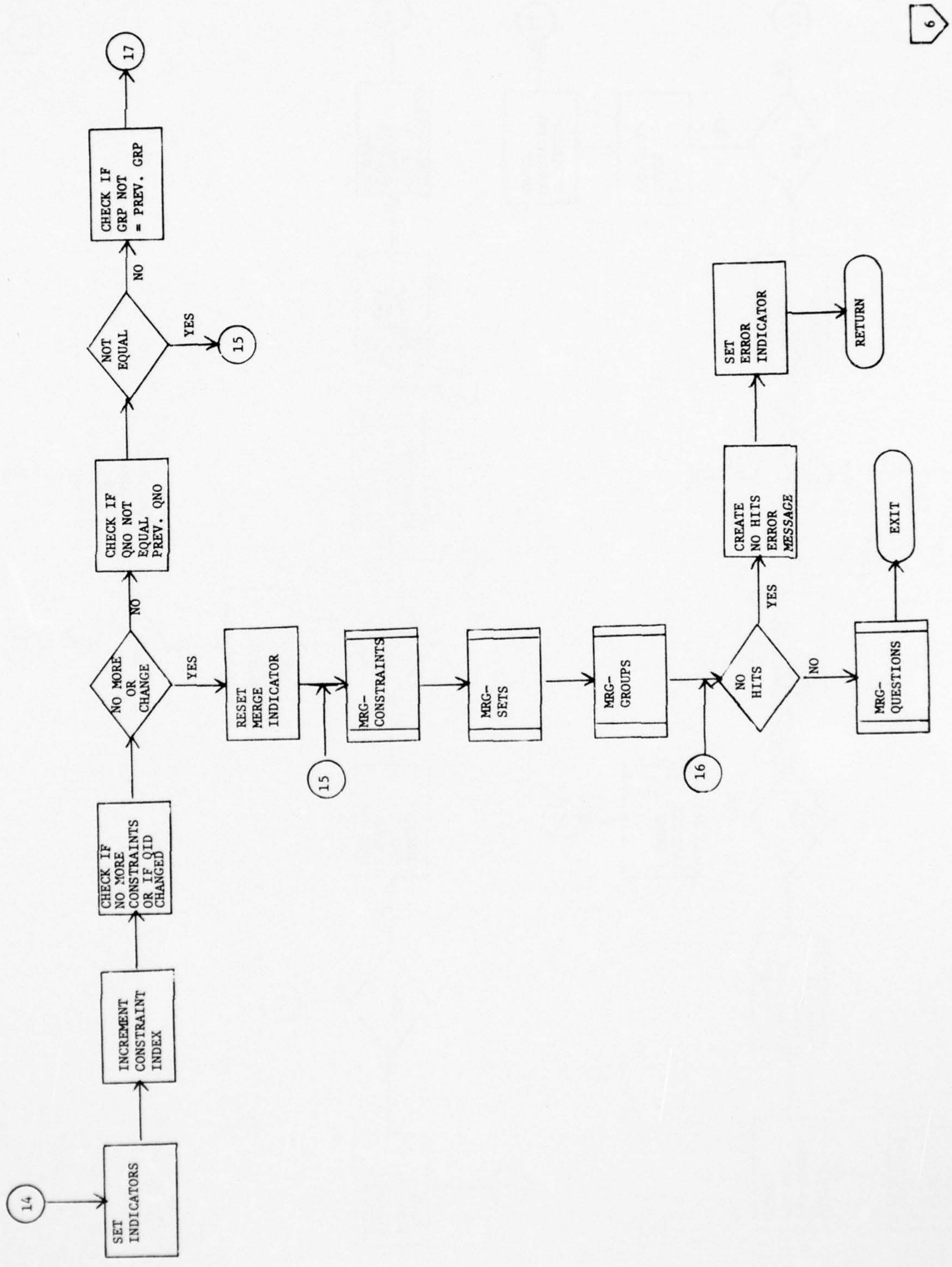

ENTRY PHRM
CHECK IF FIRST CALL TO PHRM FOR QUERY
FIRST CALL
YES
NO
PHRM-DET (DETERMINE IF PHRM CAN BE USED)
CAN BE USED
YES
NO
OPEN CORRECT INV-MF
OPEN RDF IF APPLICABLE
PRELIM PROCESSING FOR NEW QUERY
2
SET INDICATORS
RETURN
1
CHECK IF RETURN FROM NO HITS ERROR
NO HITS RETURN
YES
SET INDICATORS
2
CHECK IF RETURN FROM INVALID RDF KEY ERROR
INVALID KEY RETURN
YES
79
NO
1

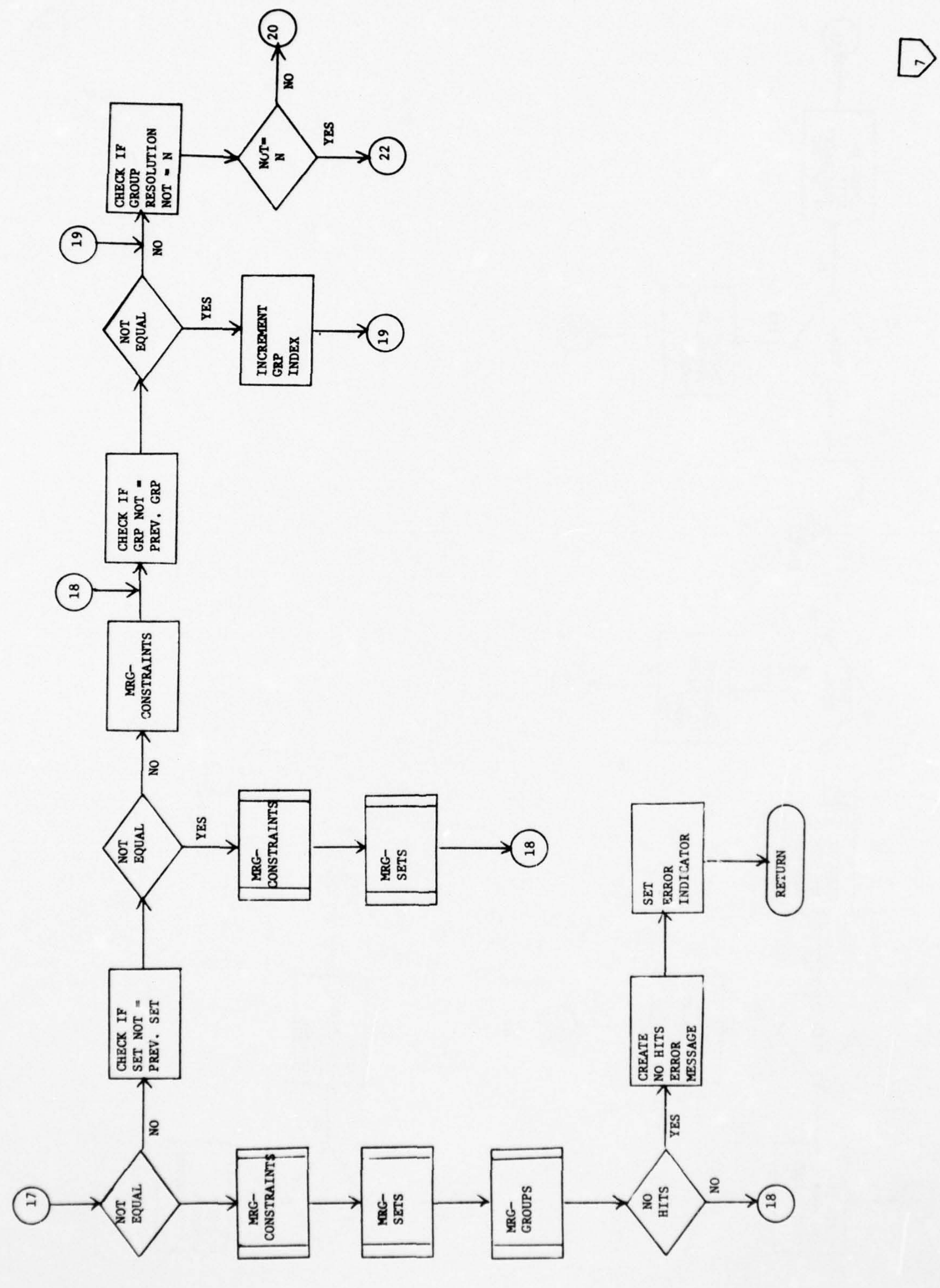

2
CHECK IF ANY MORE CONSTRAINTS
MORE
YES
NO
CHECK IF NEW QST RESOLUTION IS C,P, OR N
NOT "N"
YES
NO
FIND HITS (FOR QUESTION)
2
BYPASS QUESTION
2
CHECK IF ANY HITS STILL TO BE WRITTEN
SOME LEFT
YES
NO
WRITE HIT-FILE
RETURN
RETURN
2

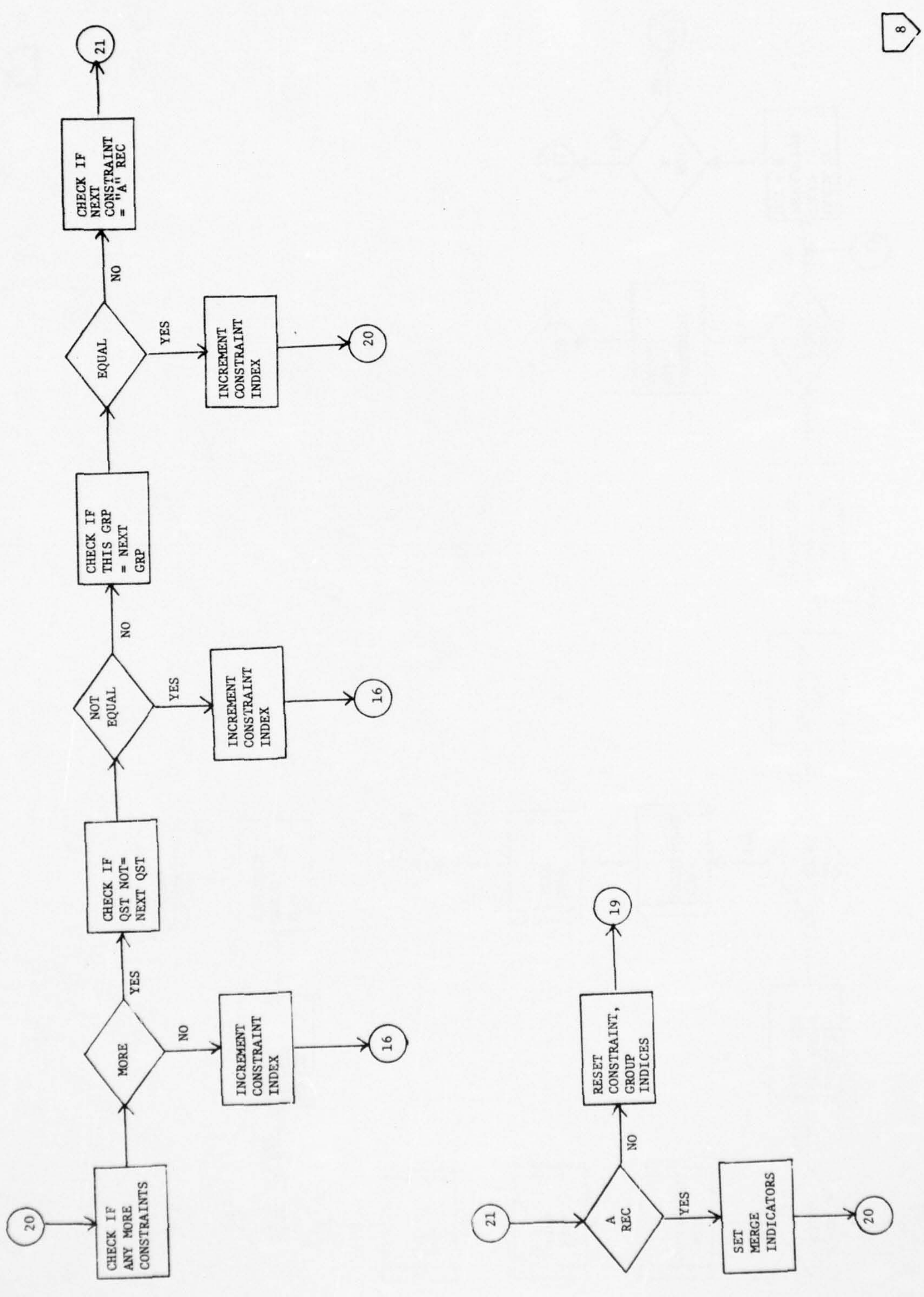

PHRM-DET
PRELIM PROCESSING FOR PHRM-DET
FIND * REC WHICH CORR. TO FILE-ID
FOUND
YES
NO
RETURN
READ QCF
EOF
YES
RETURN
NO
SET INDICATORS
3
* REC
NO
4
SET UP NEXT-FILEID
6
4
L-Z REC
YES
STORE CONSTRAINT
NO
A REC
YES
STORE CONSTRAINT
5
5
READ QCF
EOF
YES
SET INDICATORS
6
NO
3
GRP NOT=PREV. GRP
6
NO
QST NOT=PREV
YES
6
NO
7
TEST IF CONSTRAINT IS INVERTED
INVERTED
YES
5
NO
SET INDICATOR
5

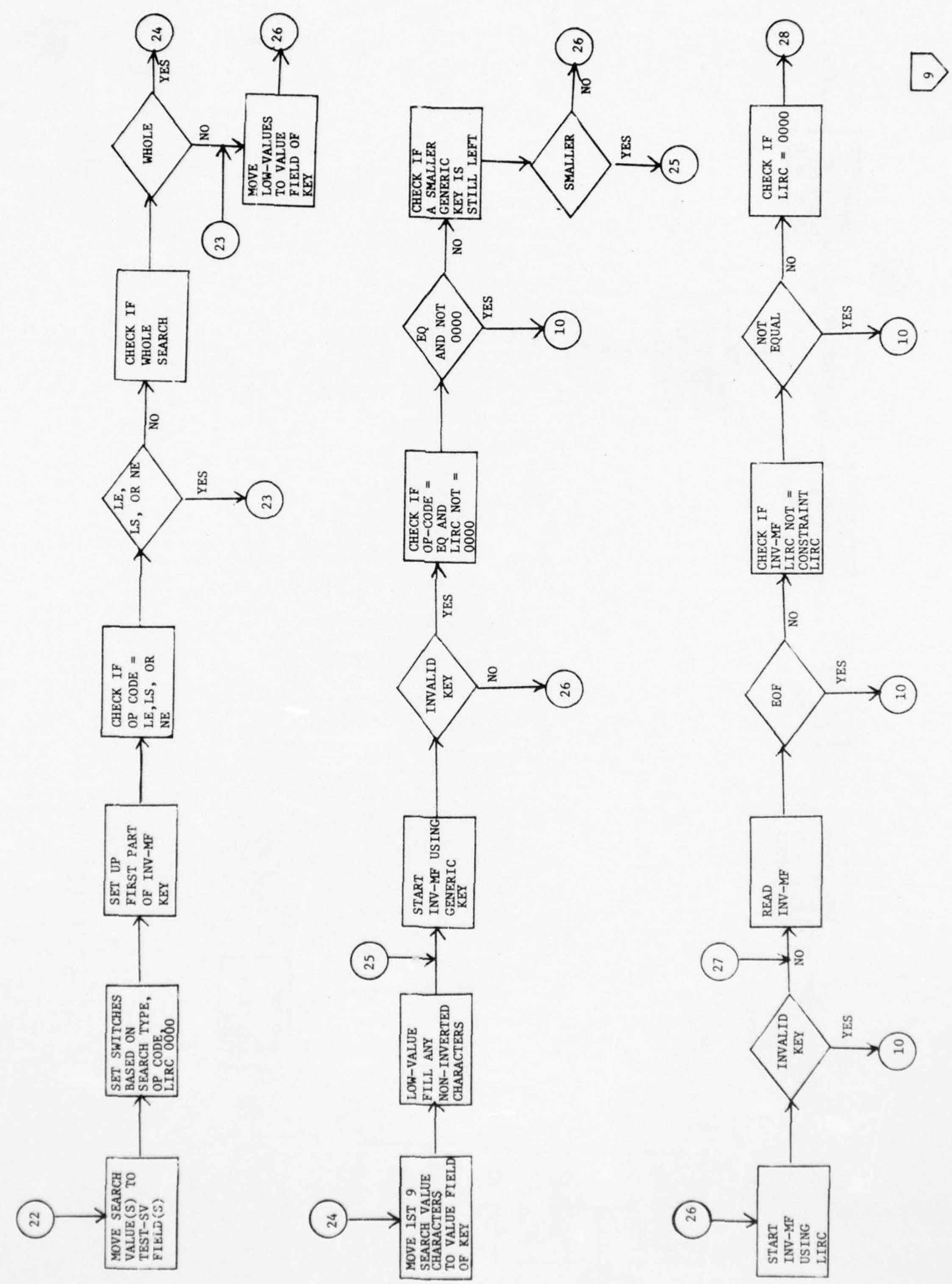

6
CHECK IF ALL CONSTRAINTS IN GROUP ARE INVERTED
ALL INVERTED
YES
NO
MOVE "C" TO GROUP RESOLUTION
MOVE "N" TO GROUP RESOLUTION
CHECK IF QUESTION RESOLUTION IS "C" OR SPACE
CHECK IF QUESTION RESOLUTION IS "C" OR "P"
"C" OR SPACE
"C" OR "P"
MOVE "C" TO QUESTION RESOLUTION
MOVE "P" TO QUESTION RESOLUTION
MOVE "N" TO QUESTION RESOLUTION
8
9
CHECK IF EOF ON QCF OR NEXT-FILEID FOUND
EOF OR NEXT
EXIT
RESET INDICATORS
7
CHECK IF NEXT CONSTRAINT IS NEW QST
NEW QST
QUESTION RESOLUTION = N
SET INDICATORS
4

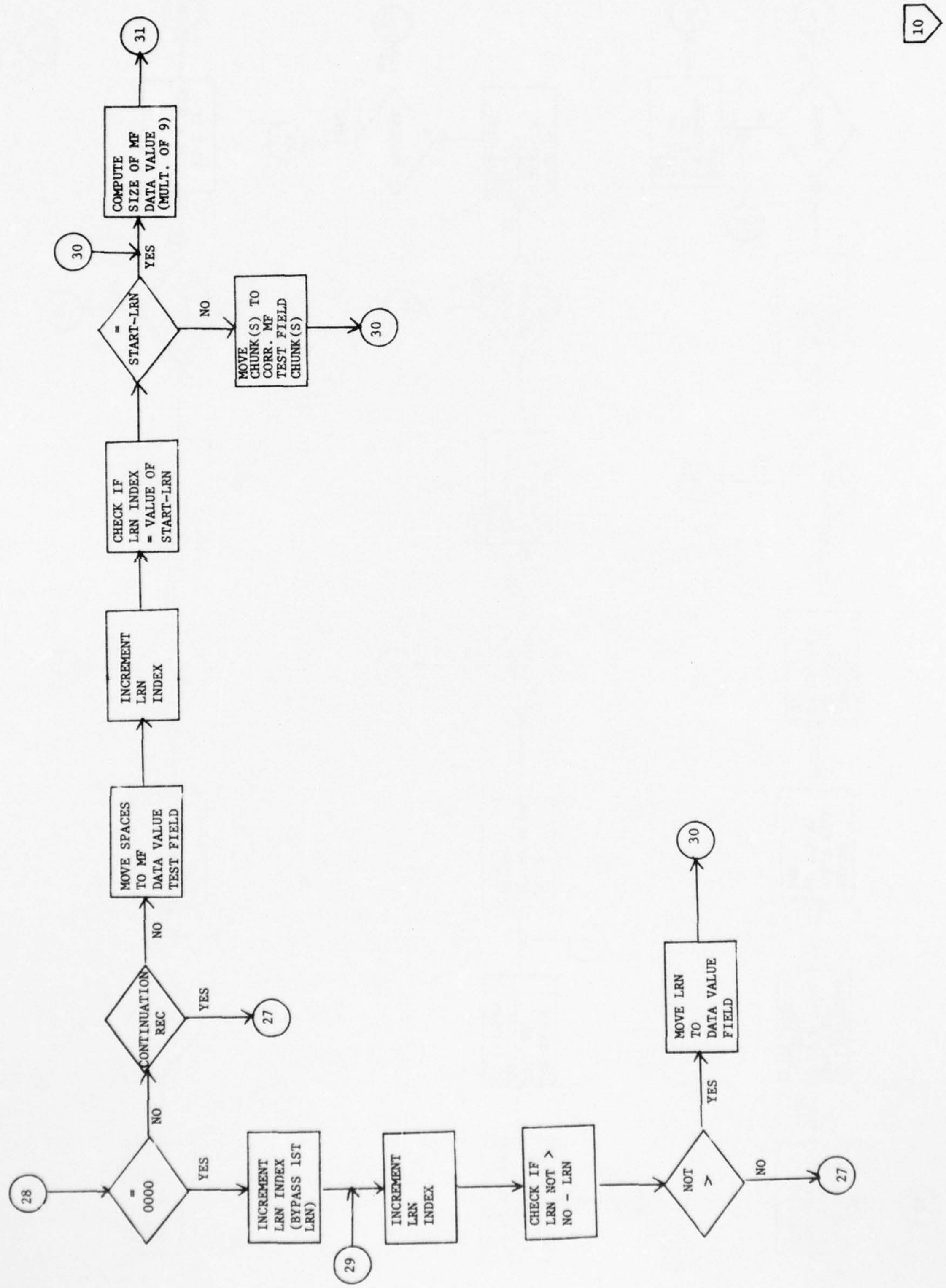

FIND HITS (FOR QUESTION)
PRELIM. PROCESSING FOR NEW QUERY
CHECK IF FIRST CONSTRAINT IS "A" REC
A REC
YES
NO
ANALYZE GROUP
19
SET MERGE INDICATOR
10
LAST CONSTRAINT
YES
13
NO
CHECK IF NEXT CONSTRAINT ="A" REC
A REC
NO
11
YES
SET MERGE INDICATOR
INCREMENT CONSTRAINT INDEX
10
11
CHECK NEXT QST EQUALS LAST QST
EQUAL
NO
12
YES
CHECK IF NEXT GRP EQUALS LAST GRP
EQUAL
YES
CHECK IF GROUP CONDITION IS "OR"
OR
NO
13
YES
INCREMENT CONSTRAINT INDEX
22
12
NO
ANALYZE GROUP
13
SORT MERGE HIT CANDIDATES VIA TABLE SORT
CLOSE SCRATCH FILES IN OPPOSITE ORDER OF OPENING
14
5

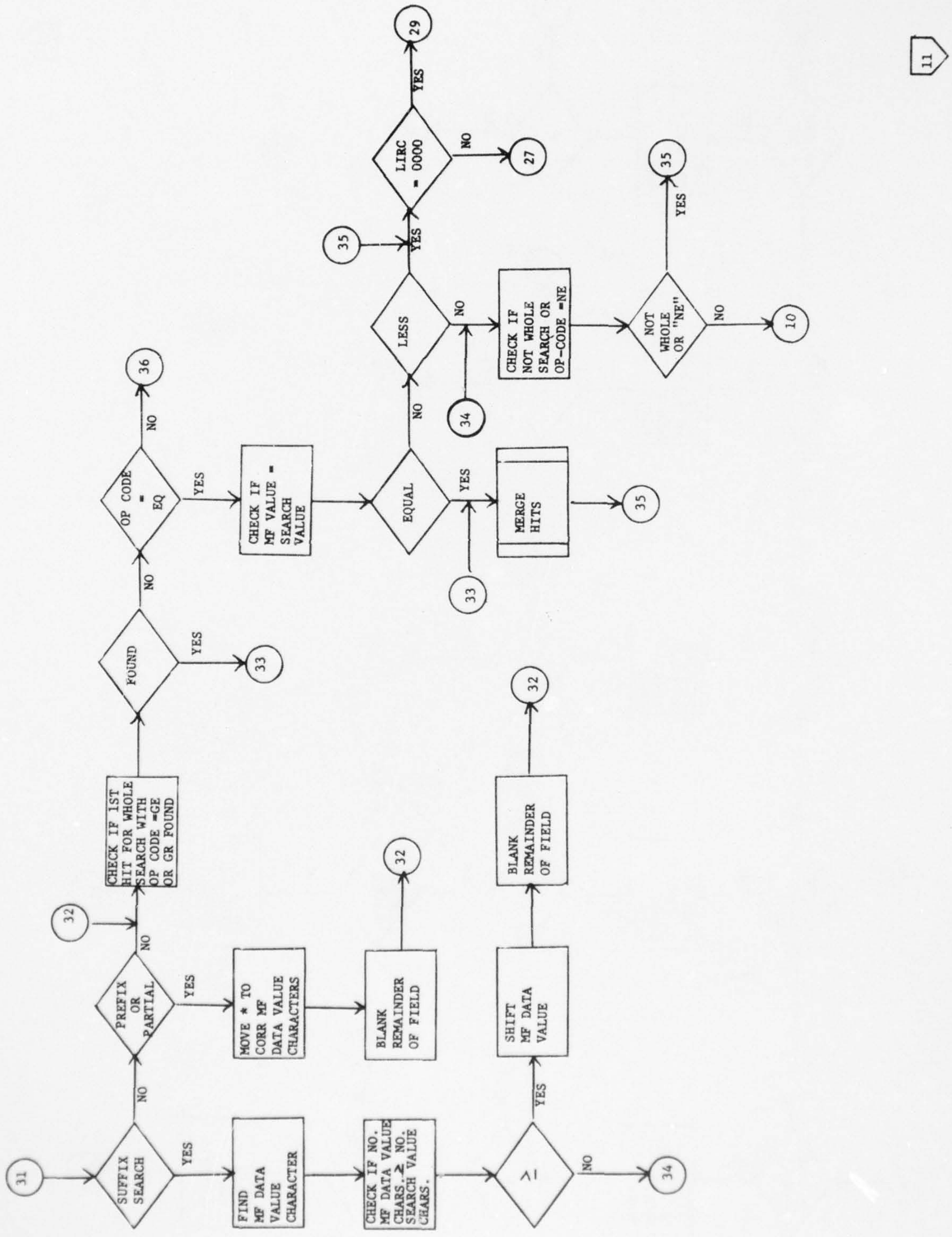

14
SET INDICATORS
INCREMENT CONSTRAINT INDEX
CHECK IF NO MORE CONSTRAINTS OR IF QID CHANGED
NO MORE OR CHANGE
NO
CHECK IF QNO NOT EQUAL PREV. QNO
NOT EQUAL
NO
CHECK IF GRP NOT = PREV. GRP
17
YES
15
YES
RESET MERGE INDICATOR
15
MRG-CONSTRAINTS
MRG-SETS
MRG-GROUPS
16
NO HITS
YES
CREATE NO HITS ERROR MESSAGE
SET ERROR INDICATOR
RETURN
NO
MRG-QUESTIONS
EXIT
6


17
NOT EQUAL
NO
CHECK IF SET NOT = PREV. SET
NOT EQUAL
YES
MRG-CONSTRAINTS
MRG-SETS
18
NO
MRG-CONSTRAINTS
18
CHECK IF GRP NOT = PREV. GRP
NOT EQUAL
YES
INCREMENT GRP INDEX
19
NO
19
CHECK IF GROUP RESOLUTION NOT = N
NOT = N
YES
22
NO
20
MRG-CONSTRAINTS
MRG-SETS
MRG-GROUPS
NO HITS
NO
18
YES
CREATE NO HITS ERROR MESSAGE
SET ERROR INDICATOR
RETURN
7


20
CHECK IF ANY MORE CONSTRAINTS
MORE
YES
NO
INCREMENT CONSTRAINT INDEX
16
CHECK IF QST NOT= NEXT QST
NOT EQUAL
YES
NO
INCREMENT CONSTRAINT INDEX
16
CHECK IF THIS GRP = NEXT GRP
EQUAL
YES
NO
INCREMENT CONSTRAINT INDEX
20
CHECK IF NEXT CONSTRAINT = "A" REC
21
21
A REC
YES
NO
SET MERGE INDICATORS
20
RESET CONSTRAINT, GROUP INDICES
19
8


22
MOVE SEARCH VALUE(S) TO TEST-SV FIELD(S)
SET SWITCHES BASED ON SEARCH TYPE, OP CODE, LIRC 0000
SET UP FIRST PART OF INV-MF KEY
CHECK IF OP CODE = LE, LS, OR NE
LE, LS, OR NE
YES
23
NO
CHECK IF WHOLE SEARCH
WHOLE
YES
24
NO
23
MOVE LOW-VALUES TO VALUE FIELD OF KEY
26
24
MOVE 1ST 9 SEARCH VALUE CHARACTERS TO VALUE FIELD OF KEY
LOW-VALUE FILL ANY NON-INVERTED CHARACTERS
25
START INV-MF USING GENERIC KEY
INVALID KEY
NO
26
YES
CHECK IF OP-CODE = EQ AND LIRC NOT = 0000
EQ AND NOT 0000
YES
10
NO
CHECK IF A SMALLER GENERIC KEY IS STILL LEFT
SMALLER
YES
25
NO
26
26
START INV-MF USING LIRC
INVALID KEY
YES
10
NO
27
READ INV-MF
EOF
YES
10
NO
CHECK IF INV-MF LIRC NOT = CONSTRAINT LIRC
NOT EQUAL
YES
10
NO
CHECK IF LIRC = 0000
28
9

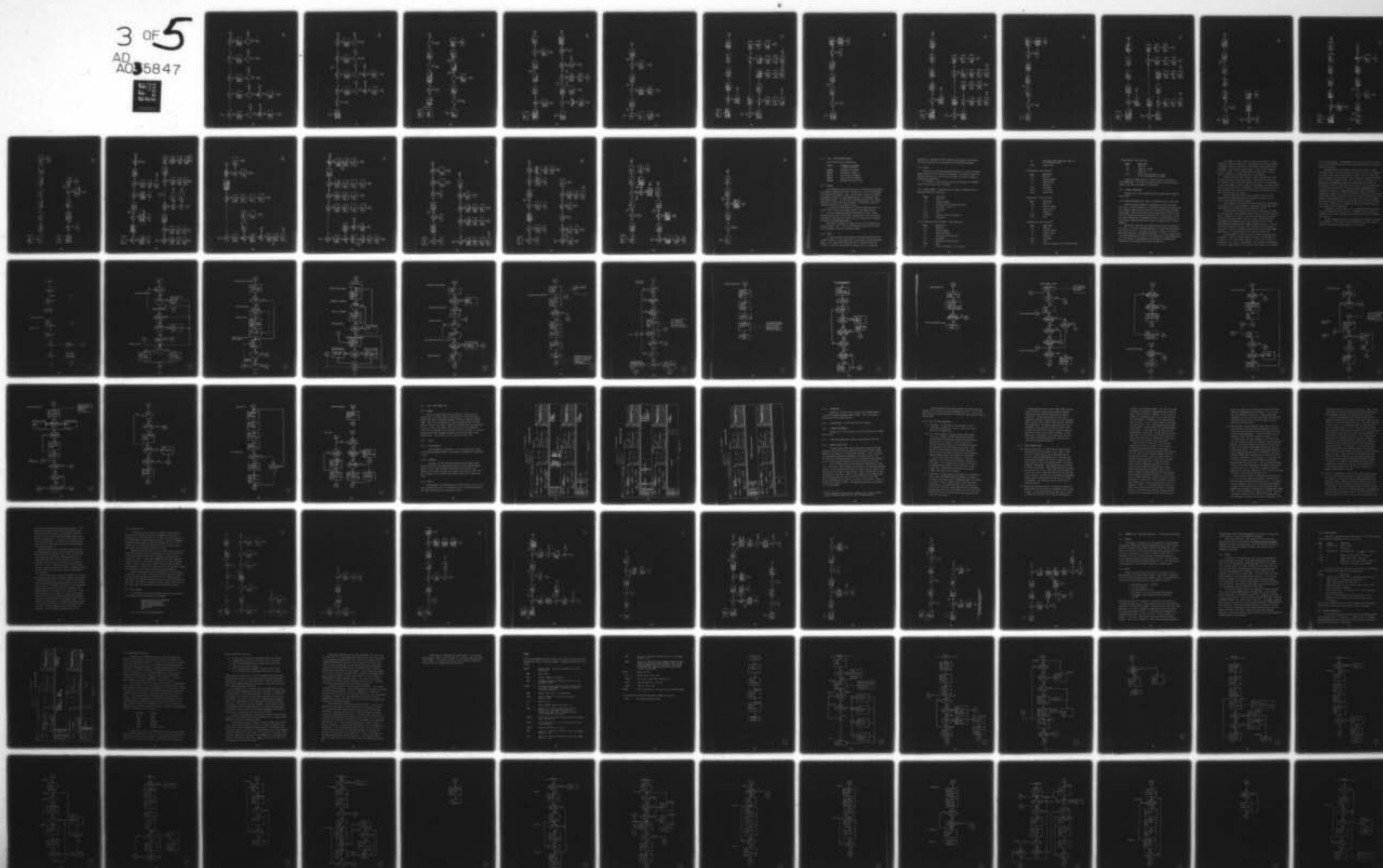

28
=
0000
YES
NO
CONTINUATION
REC
YES
27
NO
MOVE SPACES
TO MF
DATA VALUE
TEST FIELD
INCREMENT
LRN
INDEX
CHECK IF
LRN INDEX
= VALUE OF
START-LRN
=
START-LRN
NO
MOVE
CHUNK(S) TO
CORR. MF
TEST FIELD
CHUNK(S)
30
YES
30
COMPUTE
SIZE OF MF
DATA VALUE
(MULT. OF 9)
31
INCREMENT
LRN INDEX
(BYPASS 1ST
LRN)
29
INCREMENT
LRN
INDEX
CHECK IF
LRN NOT >
NO - LRN
NOT
>
YES
MOVE LRN
TO
DATA VALUE
FIELD
30
NO
27
10

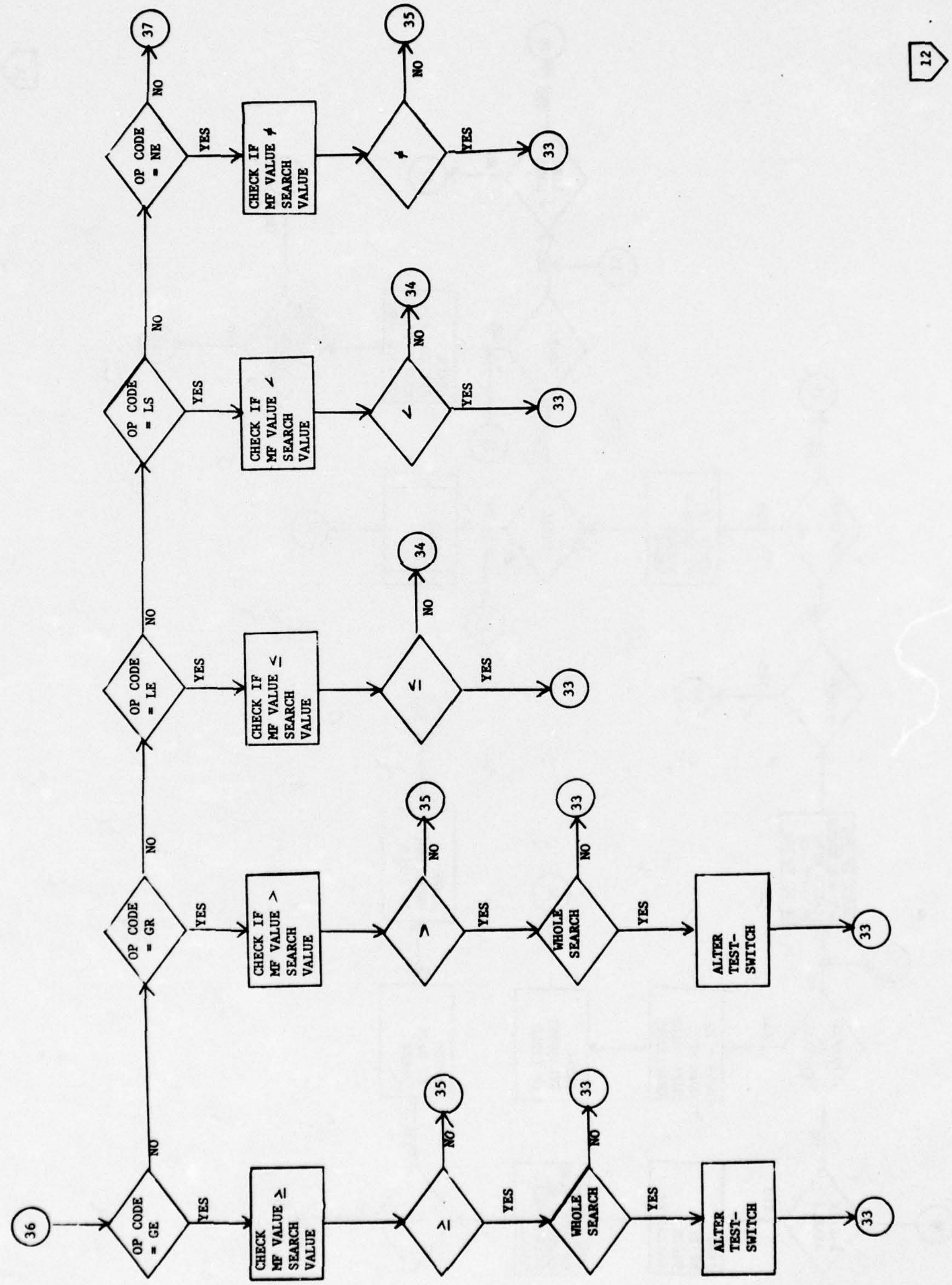

31
SUFFIX SEARCH
YES
FIND MF DATA VALUE CHARACTER
CHECK IF NO. MF DATA VALUE CHARS. ≥ NO. SEARCH VALUE CHARS.
≥
NO
34
YES
SHIFT MF DATA VALUE
BLANK REMAINDER OF FIELD
32
NO
PREFIX OR PARTIAL
YES
MOVE * TO CORR MF DATA VALUE CHARACTERS
BLANK REMAINDER OF FIELD
32
NO
32
CHECK IF 1ST HIT FOR WHOLE SEARCH WITH OP CODE =GE OR GR FOUND
FOUND
YES
33
NO
OP CODE = EQ
NO
36
YES
CHECK IF MF VALUE = SEARCH VALUE
EQUAL
YES
MERGE HITS
35
33
NO
LESS
34
YES
35
LIRC = 0000
YES
29
NO
27
NO
CHECK IF NOT WHOLE SEARCH OR OP-CODE =NE
NOT WHOLE OR "NE"
YES
35
NO
10
11

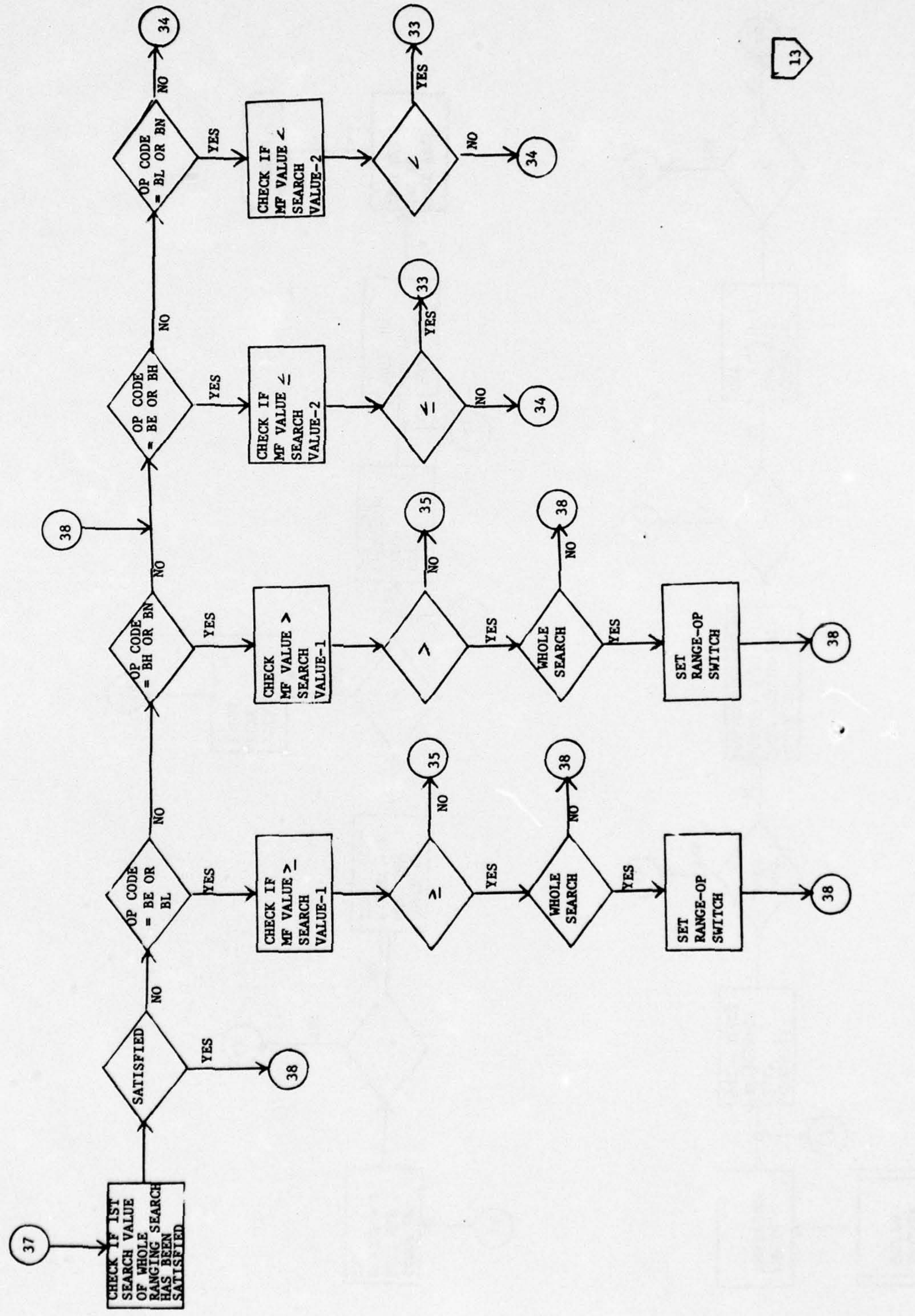
AD-A035 847   DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC   F/G 9/2
NAVY LOGISTICS INFORMATION SHARING (NAVLIS) PROJECT.(U)
JUN 76

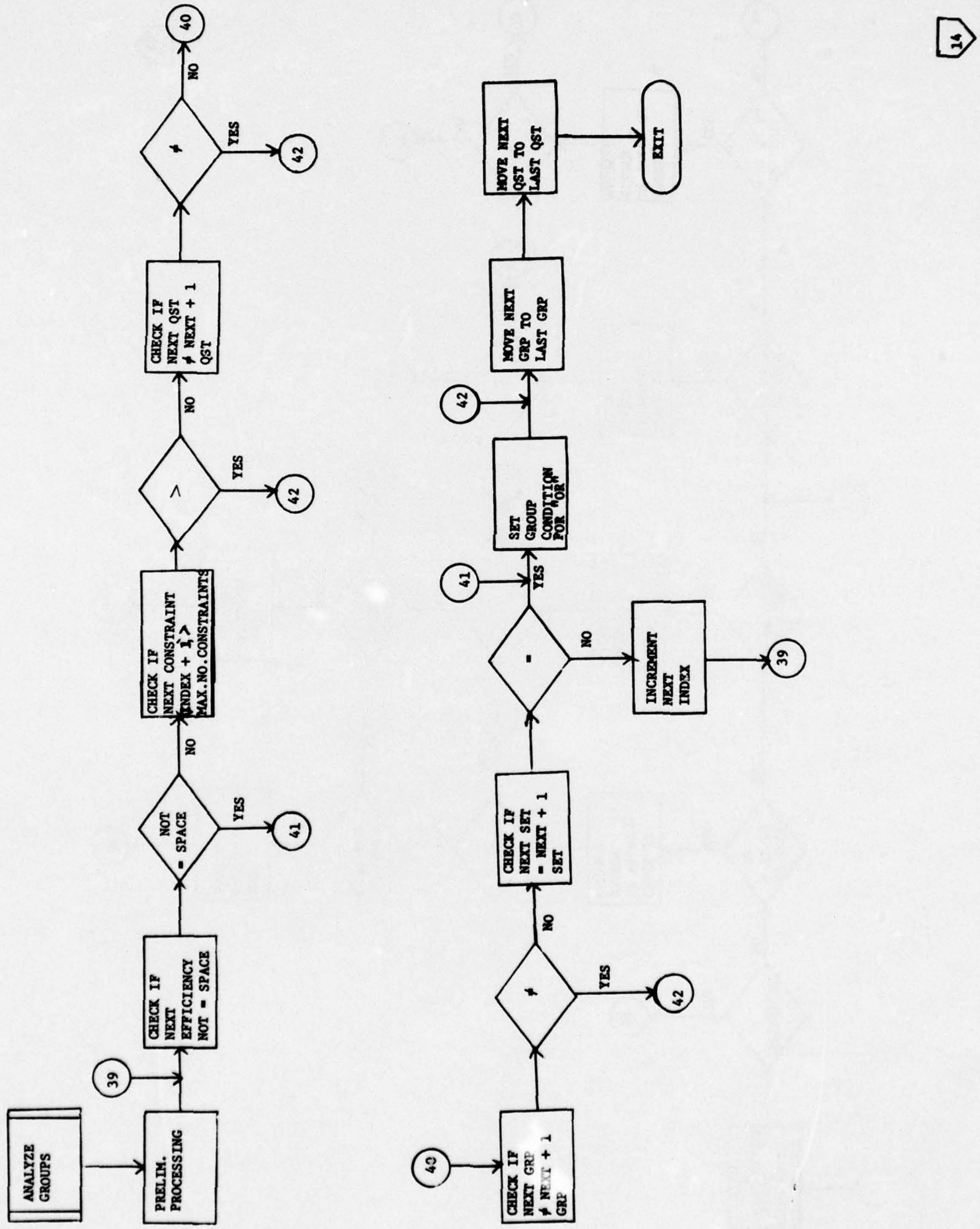
UNCLASSIFIED   DTNSRDC-76-0120   NL

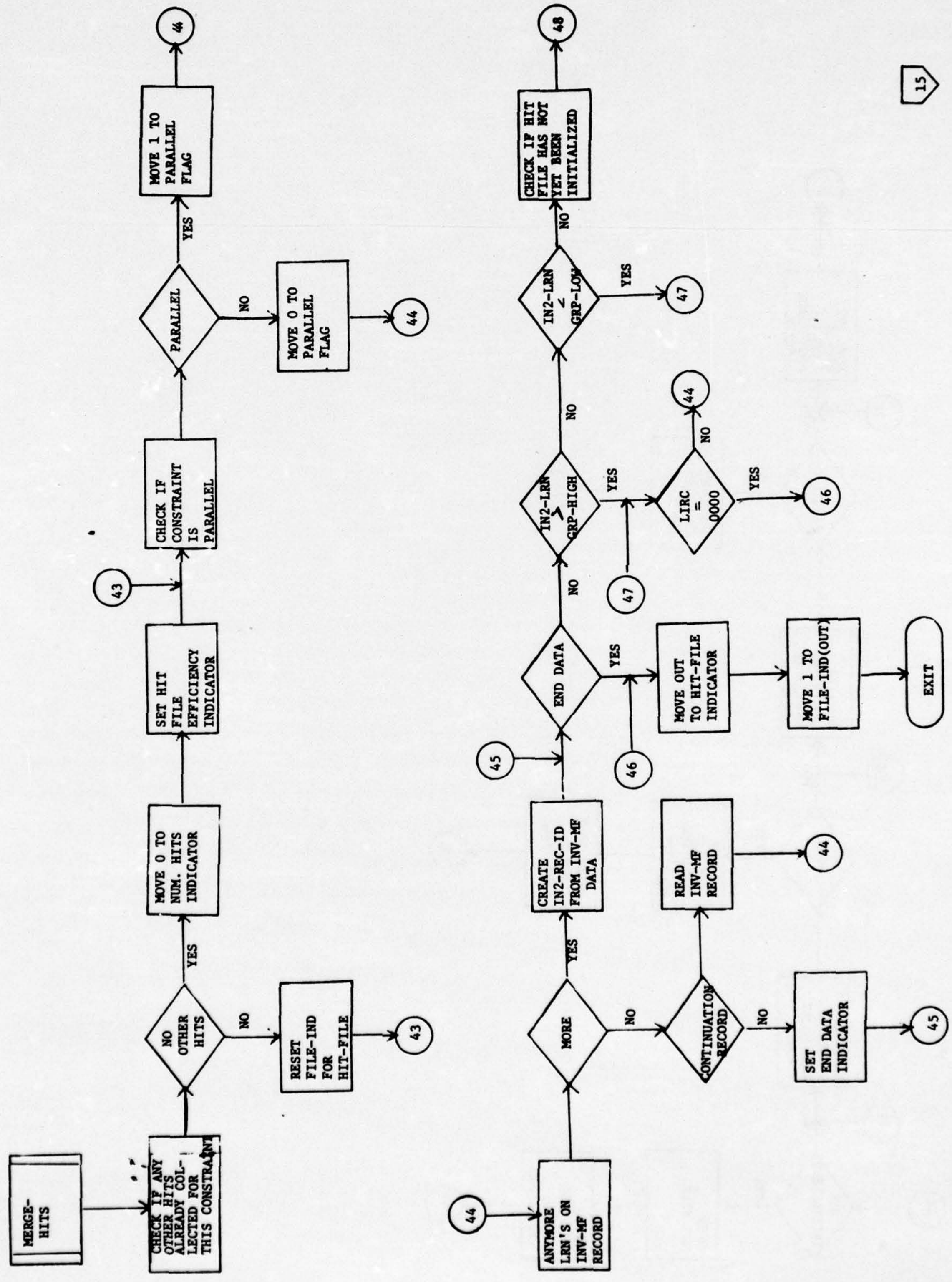
3 OF 5
AD
A035847

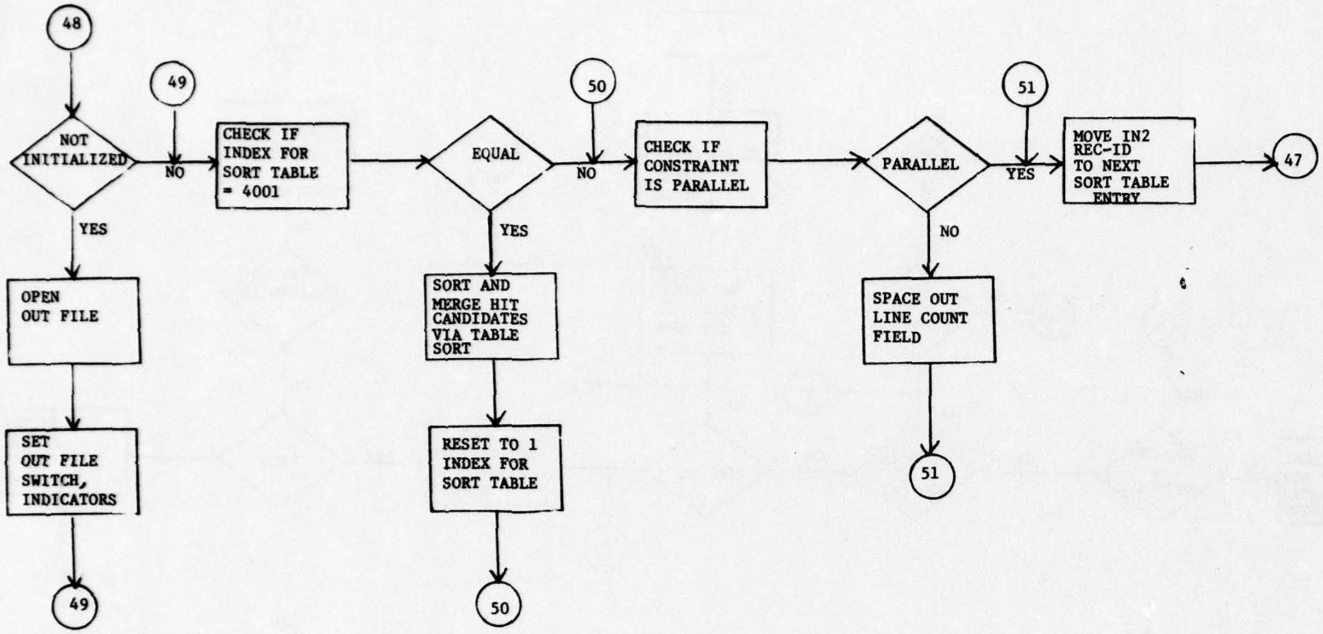


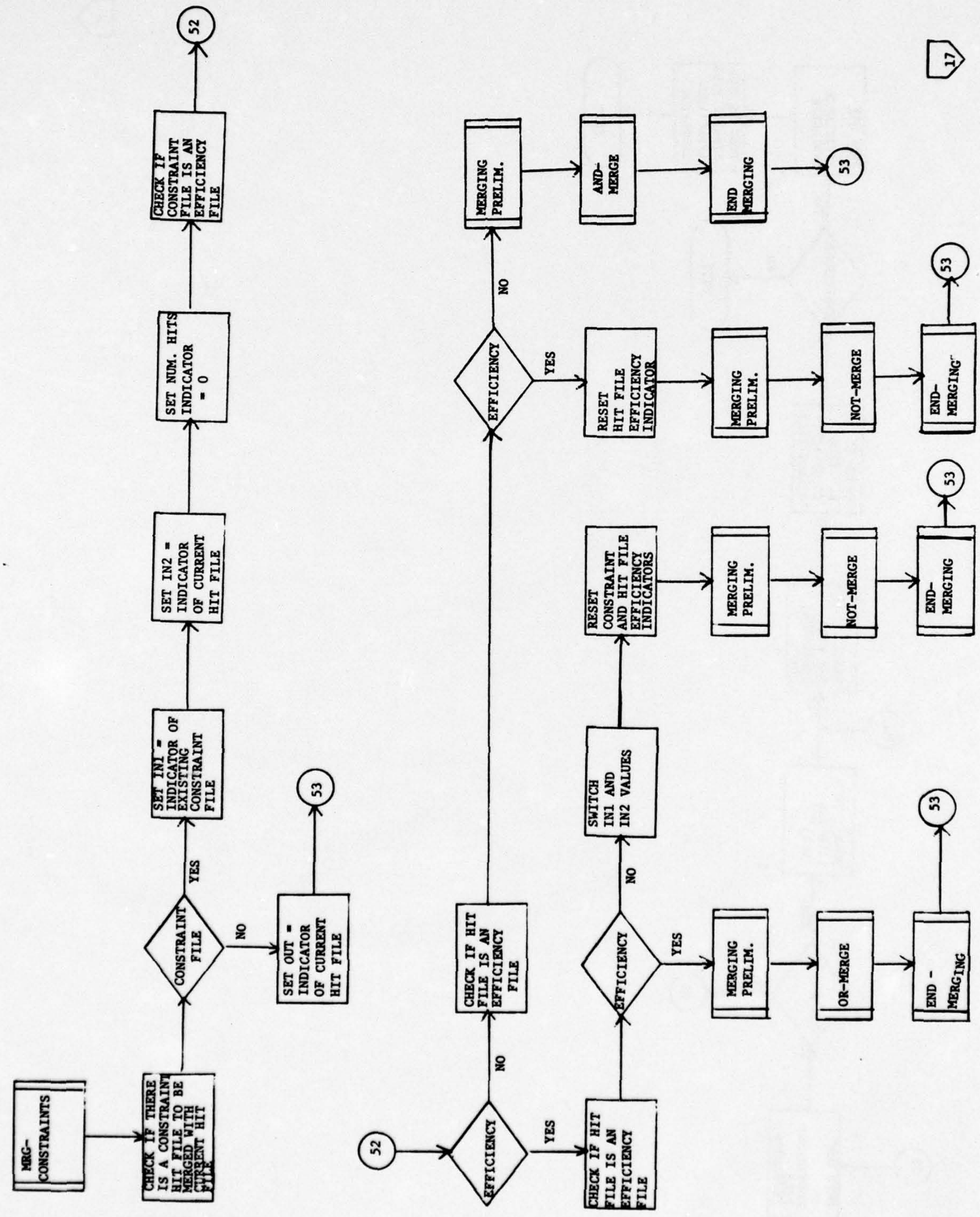

36
OP CODE = GE
CHECK MF VALUE ≥ SEARCH VALUE
≥
WHOLE SEARCH
ALTER TEST-SWITCH
OP CODE = GR
CHECK IF MF VALUE > SEARCH VALUE
>
WHOLE SEARCH
ALTER TEST-SWITCH
OP CODE = LE
CHECK IF MF VALUE ≤ SEARCH VALUE
≤
OP CODE = LS
CHECK IF MF VALUE < SEARCH VALUE
<
OP CODE = NE
CHECK IF MF VALUE ≠ SEARCH VALUE
≠
YES
NO
33
34
35
37
12


37
CHECK IF 1ST SEARCH VALUE OF WHOLE RANGING SEARCH HAS BEEN SATISFIED
SATISFIED
YES
NO
38
OP CODE = BE OR BL
CHECK IF MF VALUE ≥ SEARCH VALUE-1
≥
WHOLE SEARCH
SET RANGE-OP SWITCH
35
OP CODE = BH OR BN
CHECK MF VALUE > SEARCH VALUE-1
>
OP CODE = BE OR BH
CHECK IF MF VALUE ≤ SEARCH VALUE-2
≤
33
34
OP CODE = BL OR BN
CHECK IF MF VALUE < SEARCH VALUE-2
<
13

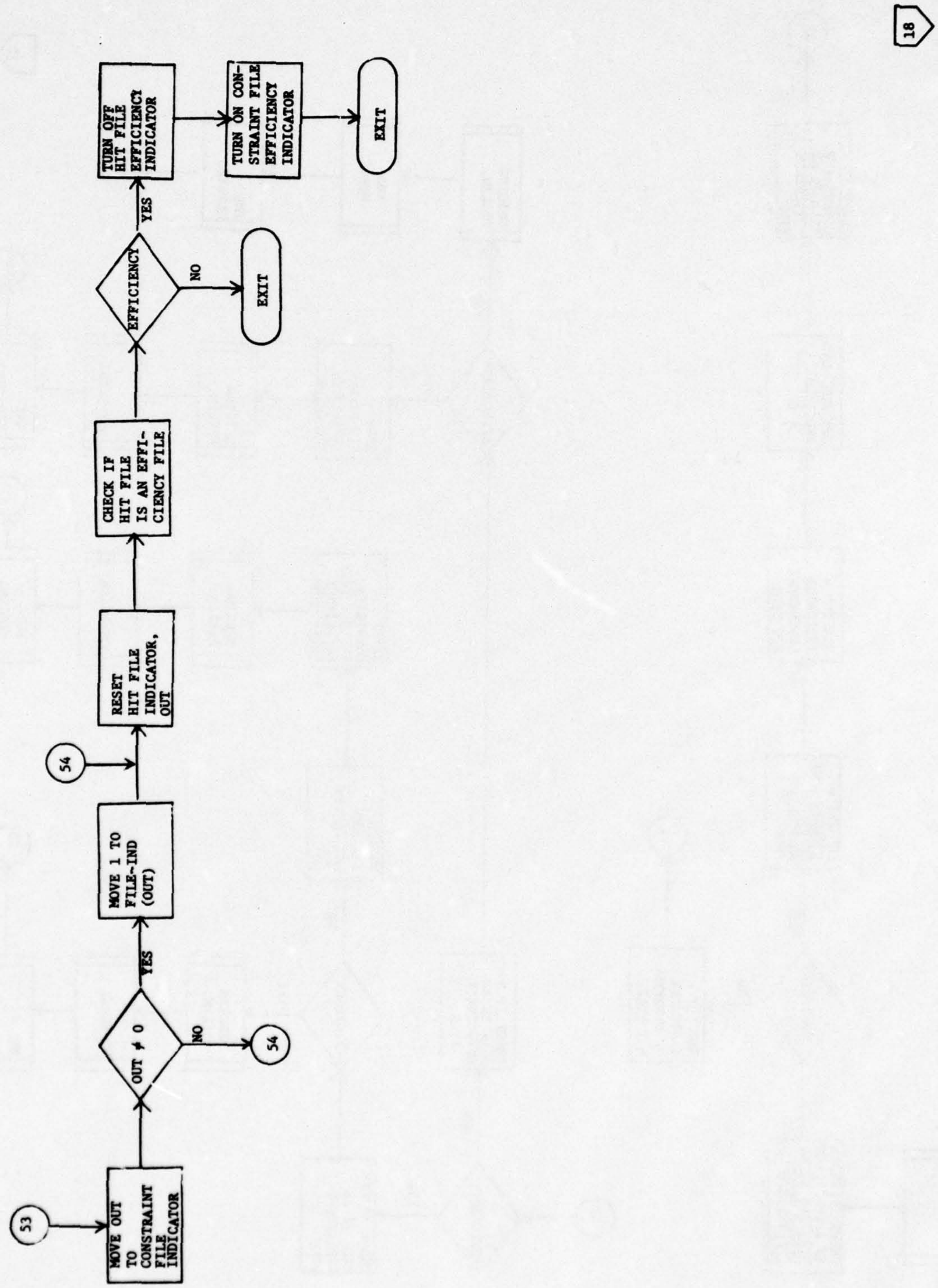

ANALYZE GROUPS
PRELIM. PROCESSING
39
CHECK IF NEXT EFFICIENCY NOT = SPACE
NOT = SPACE
YES
41
NO
CHECK IF NEXT CONSTRAINT INDEX + 1 > MAX.NO.CONSTRAINTS
>
YES
42
NO
CHECK IF NEXT QST ≠ NEXT + 1 QST
≠
YES
42
NO
40
40
CHECK IF NEXT GRP ≠ NEXT + 1 GRP
≠
YES
42
NO
CHECK IF NEXT SET = NEXT + 1 SET
=
NO
INCREMENT NEXT INDEX
39
YES
41
SET GROUP CONDITION FOR "OR"
42
MOVE NEXT GRP TO LAST GRP
MOVE NEXT QST TO LAST QST
EXIT
14

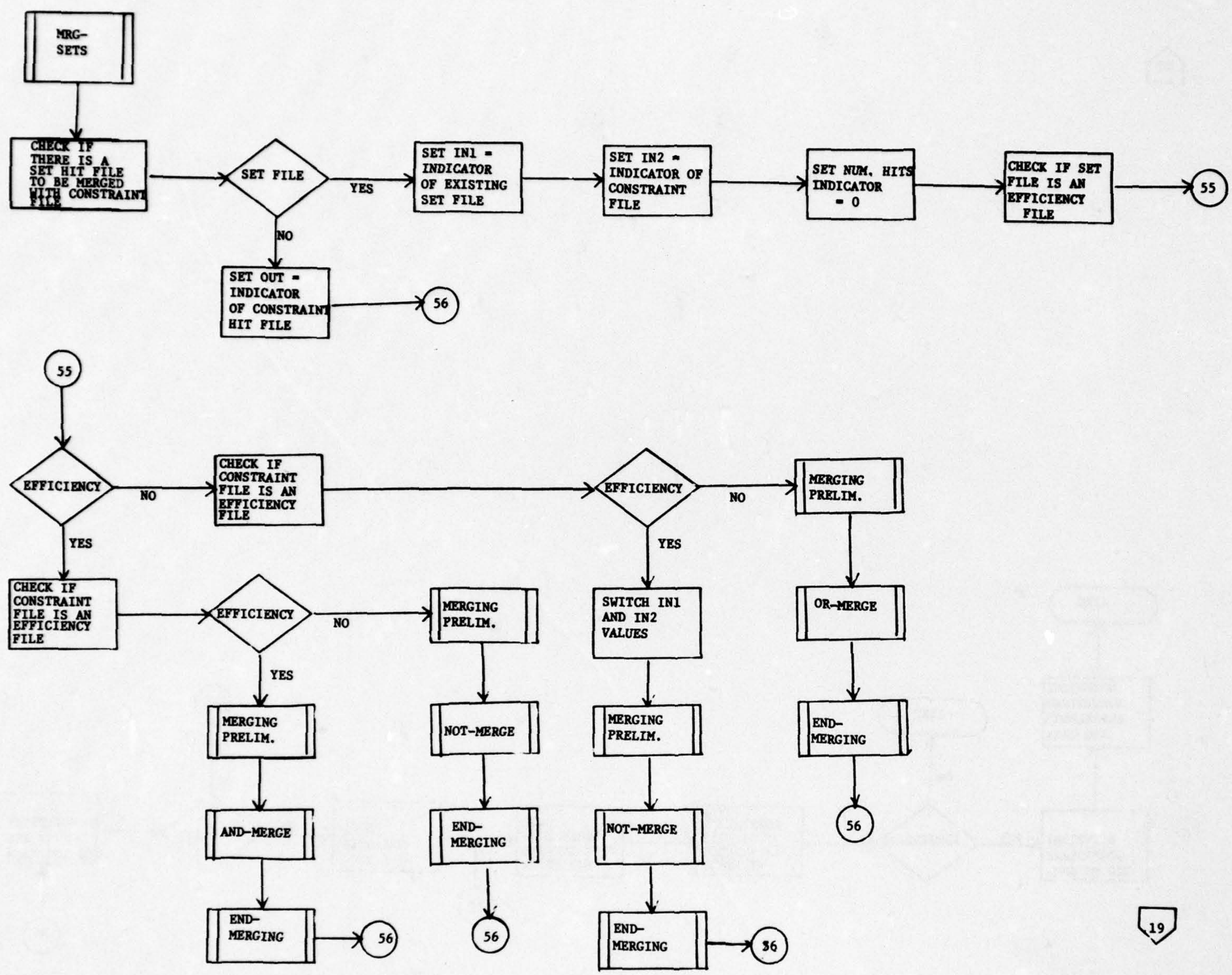

MERGE-HITS
CHECK IF ANY OTHER HITS ALREADY COL-LECTED FOR THIS CONSTRAINT
NO OTHER HITS
YES
NO
RESET FILE-IND FOR HIT-FILE
43
MOVE 0 TO NUM. HITS INDICATOR
SET HIT FILE EFFICIENCY INDICATOR
43
CHECK IF CONSTRAINT IS PARALLEL
PARALLEL
YES
NO
MOVE 1 TO PARALLEL FLAG
44
MOVE 0 TO PARALLEL FLAG
44
44
ANYMORE LRN'S ON INV-MF RECORD
MORE
YES
NO
CONTINUATION RECORD
NO
SET END DATA INDICATOR
45
READ INV-MF RECORD
44
CREATE IN2-REC-ID FROM INV-MF DATA
45
END DATA
YES
NO
46
MOVE OUT TO HIT-FILE INDICATOR
MOVE 1 TO FILE-IND(OUT)
EXIT
IN2-LRN > GRP-HIGH
YES
NO
47
LIRC = 0000
YES
NO
46
44
IN2-LRN < GRP-LOW
YES
NO
47
CHECK IF HIT FILE HAS NOT YET BEEN INITIALIZED
48
15

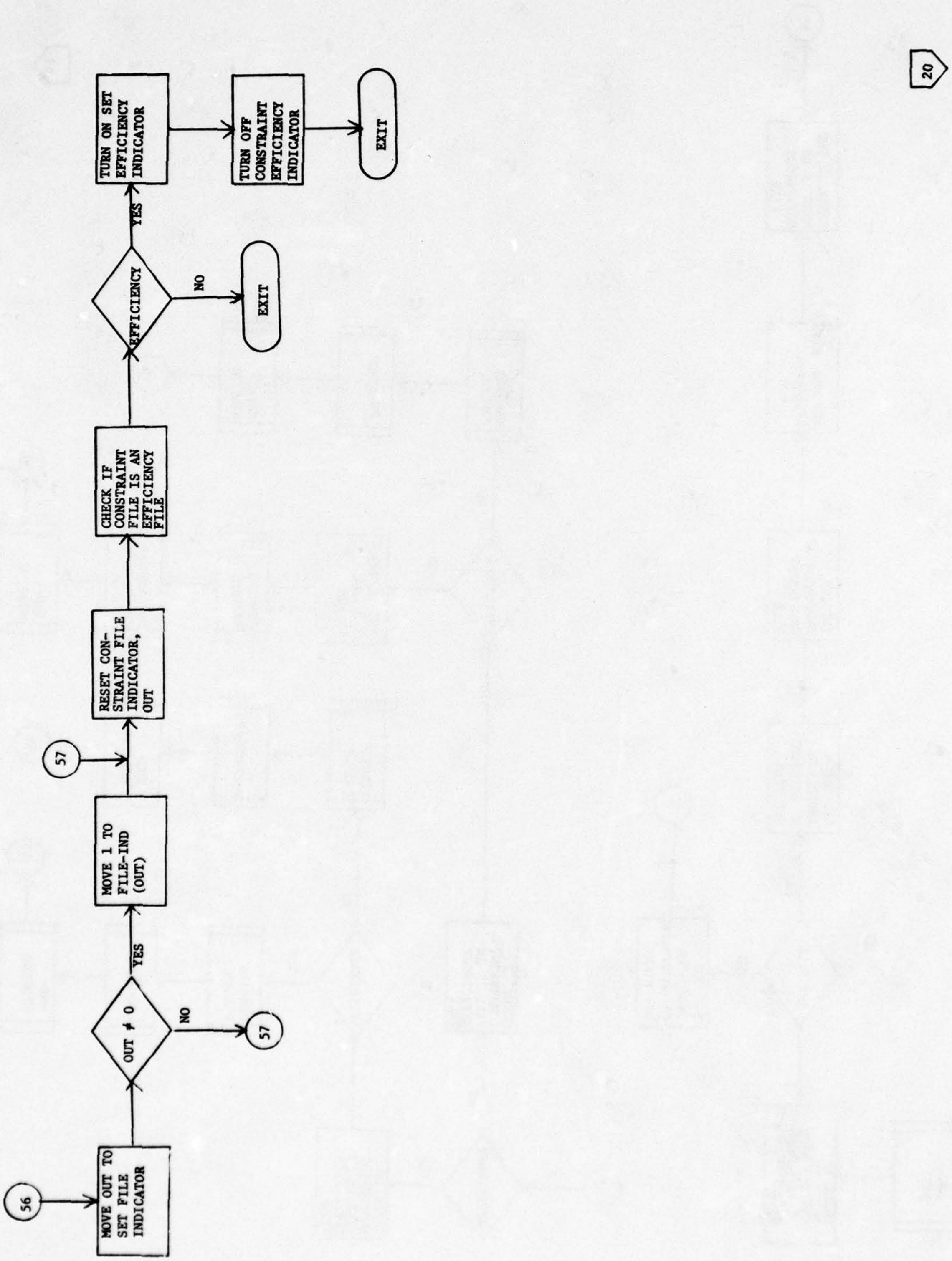

48
NOT INITIALIZED
YES
NO
49
OPEN OUT FILE
SET OUT FILE SWITCH, INDICATORS
49
CHECK IF INDEX FOR SORT TABLE = 4001
EQUAL
YES
NO
SORT AND MERGE HIT CANDIDATES VIA TABLE SORT
RESET TO 1 INDEX FOR SORT TABLE
50
50
CHECK IF CONSTRAINT IS PARALLEL
PARALLEL
YES
NO
SPACE OUT LINE COUNT FIELD
51
51
MOVE IN2 REC-ID TO NEXT SORT TABLE ENTRY
47

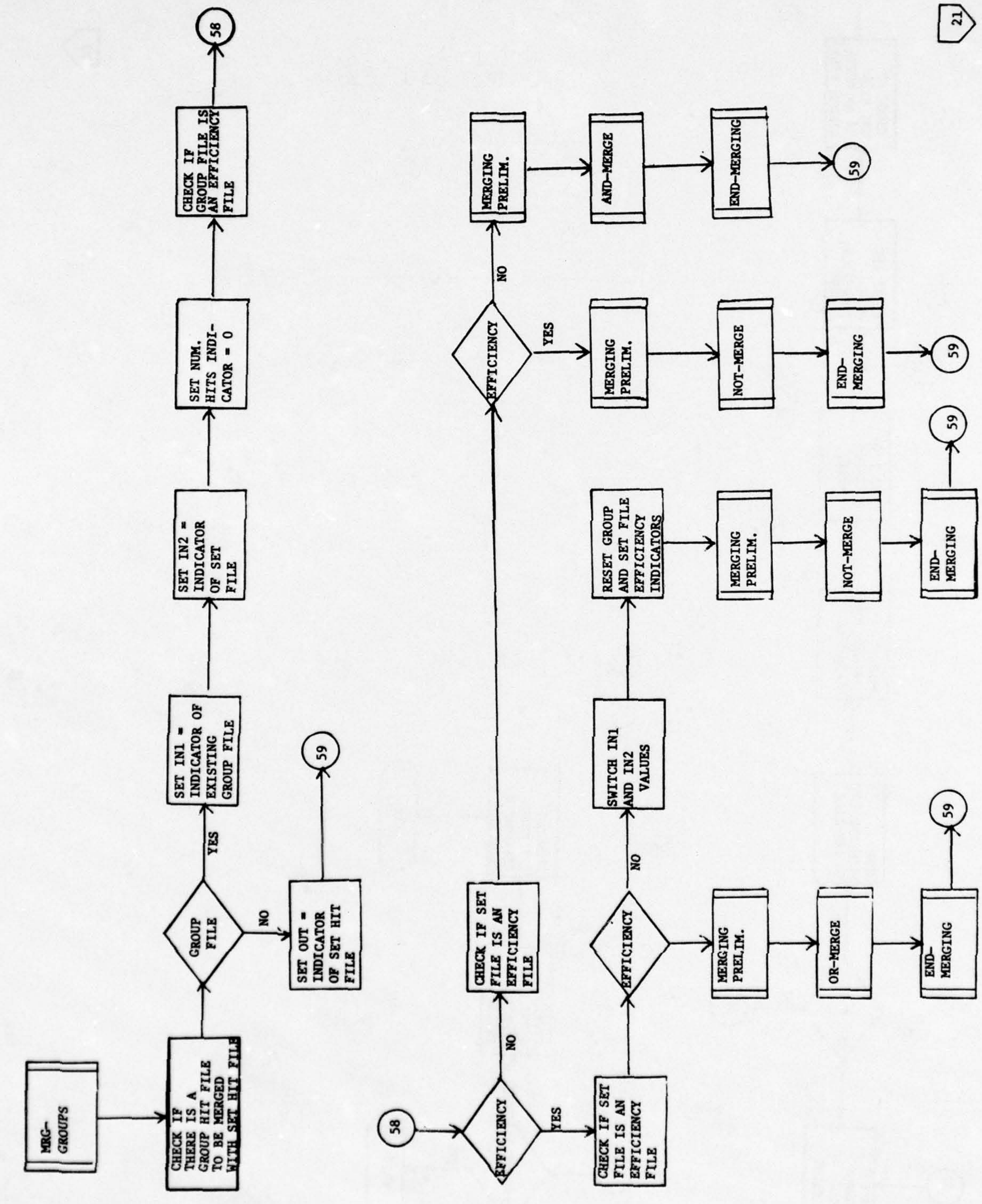


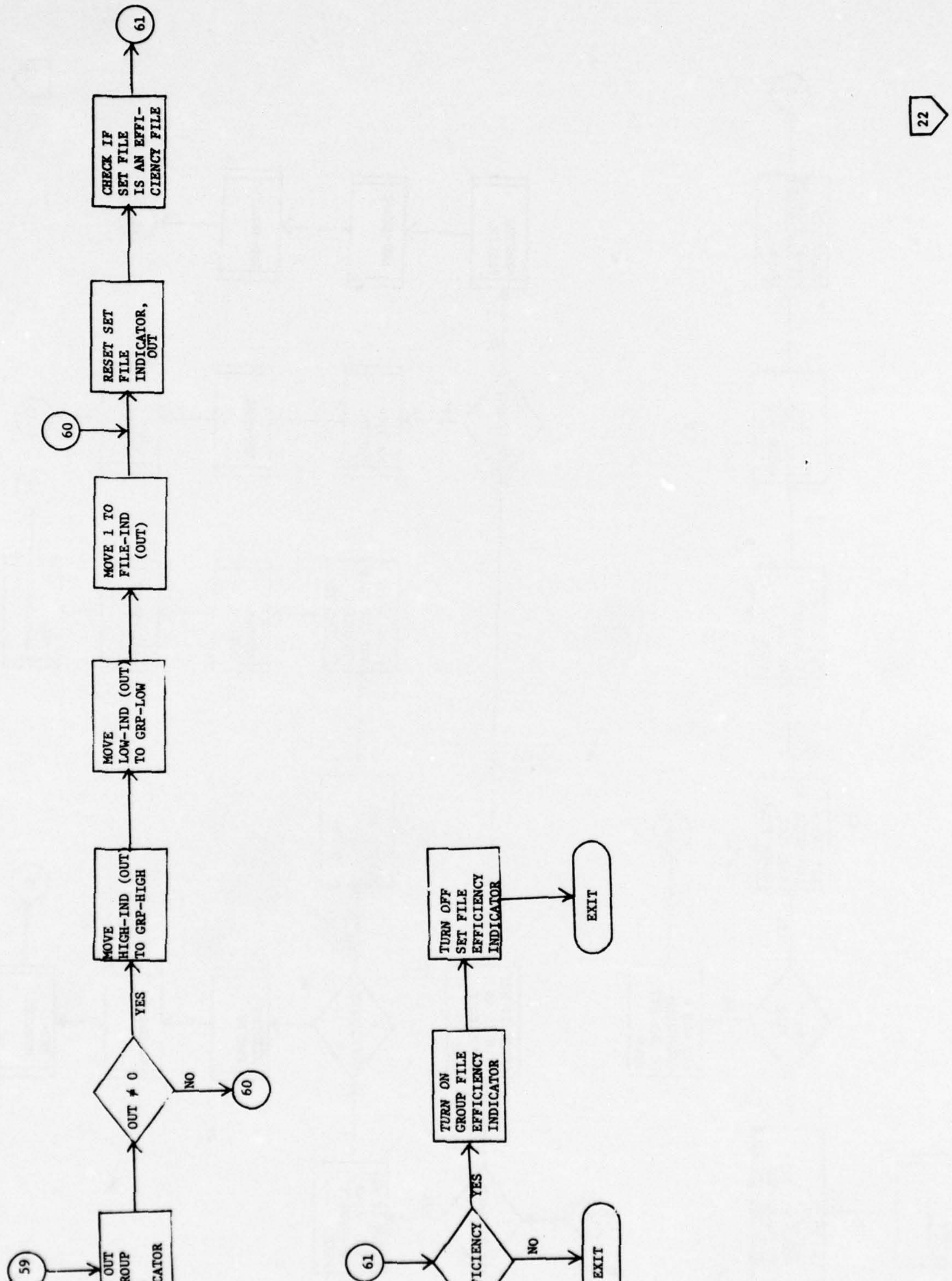
MRG-CONSTRAINTS
CHECK IF THERE IS A CONSTRAINT HIT FILE TO BE MERGED WITH CURRENT HIT FILE
CONSTRAINT FILE
YES
NO
SET OUT = INDICATOR OF CURRENT HIT FILE
53
SET IN1 = INDICATOR OF EXISTING CONSTRAINT FILE
SET IN2 = INDICATOR OF CURRENT HIT FILE
SET NUM. HITS INDICATOR = 0
CHECK IF CONSTRAINT FILE IS AN EFFICIENCY FILE
52
52
EFFICIENCY
YES
NO
CHECK IF HIT FILE IS AN EFFICIENCY FILE
CHECK IF HIT FILE IS AN EFFICIENCY FILE
EFFICIENCY
YES
NO
MERGING PRELIM.
OR-MERGE
END-MERGING
53
SWITCH IN1 AND IN2 VALUES
RESET CONSTRAINT AND HIT FILE EFFICIENCY INDICATORS
MERGING PRELIM.
NOT-MERGE
END-MERGING
53
EFFICIENCY
YES
NO
RESET HIT FILE EFFICIENCY INDICATOR
MERGING PRELIM.
NOT-MERGE
END-MERGING
53
MERGING PRELIM.
AND-MERGE
END MERGING
53
17

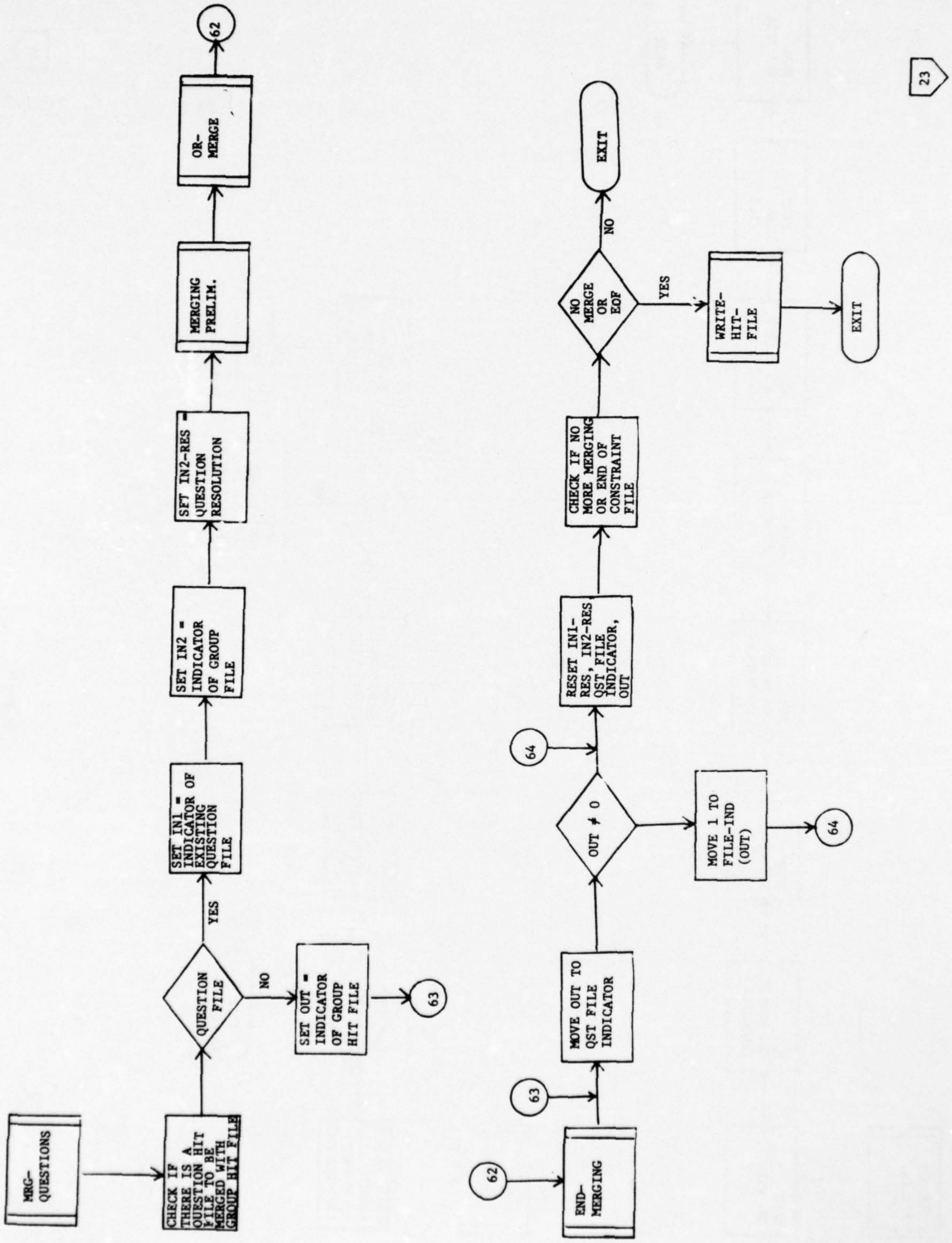

53
MOVE OUT TO CONSTRAINT FILE INDICATOR
OUT ≠ 0
NO
54
YES
MOVE 1 TO FILE-IND (OUT)
54
RESET HIT FILE INDICATOR, OUT
CHECK IF HIT FILE IS AN EFFI- CIENCY FILE
EFFICIENCY
NO
EXIT
YES
TURN OFF HIT FILE EFFICIENCY INDICATOR
TURN ON CON- STRAINT FILE EFFICIENCY INDICATOR
EXIT
18

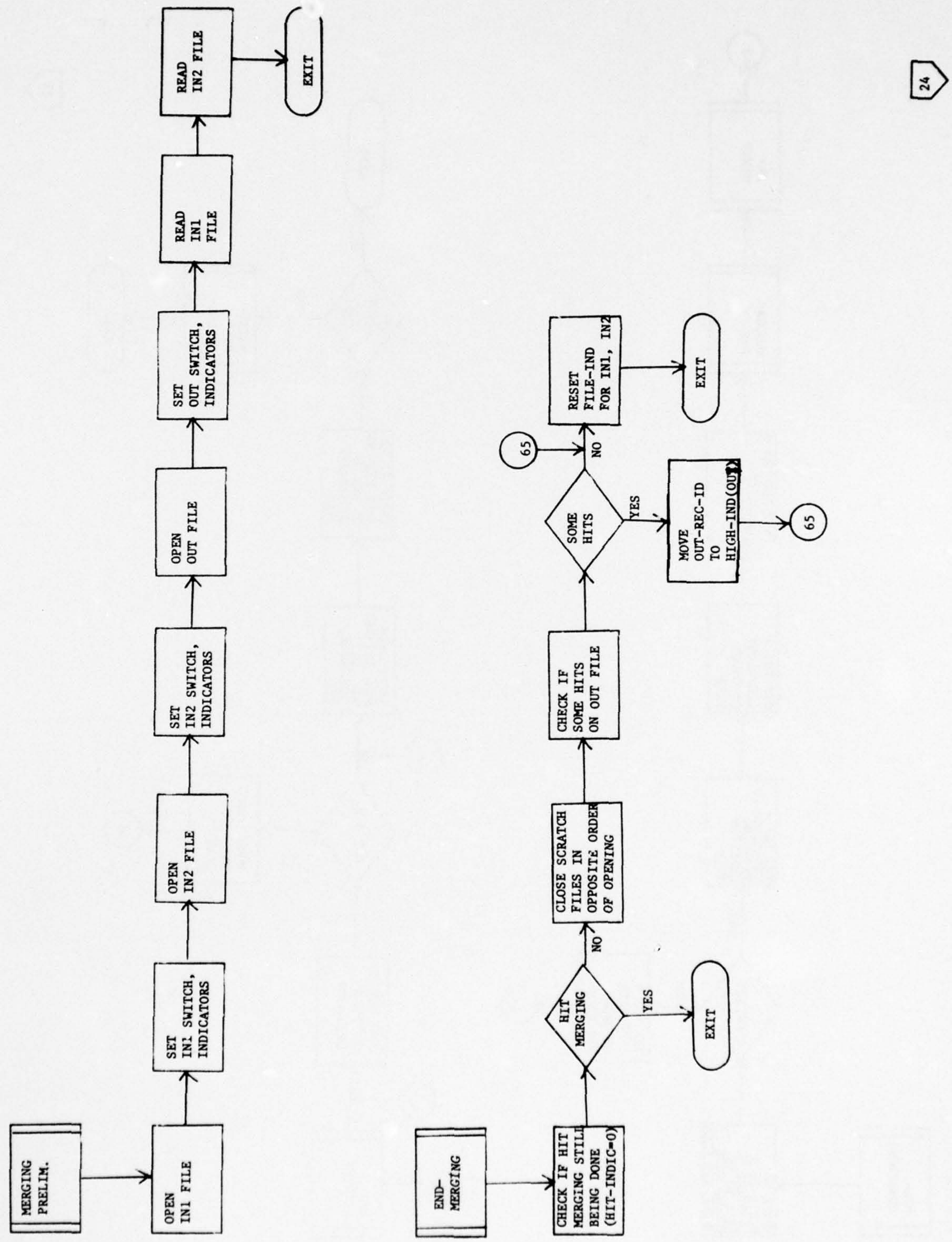

MRG-SETS
CHECK IF THERE IS A SET HIT FILE TO BE MERGED WITH CONSTRAINT FILE
SET FILE
YES
NO
SET IN1 = INDICATOR OF EXISTING SET FILE
SET IN2 = INDICATOR OF CONSTRAINT FILE
SET NUM. HITS INDICATOR = 0
CHECK IF SET FILE IS AN EFFICIENCY FILE
55
SET OUT = INDICATOR OF CONSTRAINT HIT FILE
56
55
EFFICIENCY
NO
YES
CHECK IF CONSTRAINT FILE IS AN EFFICIENCY FILE
EFFICIENCY
NO
YES
MERGING PRELIM.
OR-MERGE
END-MERGING
56
CHECK IF CONSTRAINT FILE IS AN EFFICIENCY FILE
EFFICIENCY
NO
YES
MERGING PRELIM.
NOT-MERGE
END-MERGING
56
SWITCH IN1 AND IN2 VALUES
MERGING PRELIM.
NOT-MERGE
END-MERGING
56
MERGING PRELIM.
AND-MERGE
END-MERGING
56
19

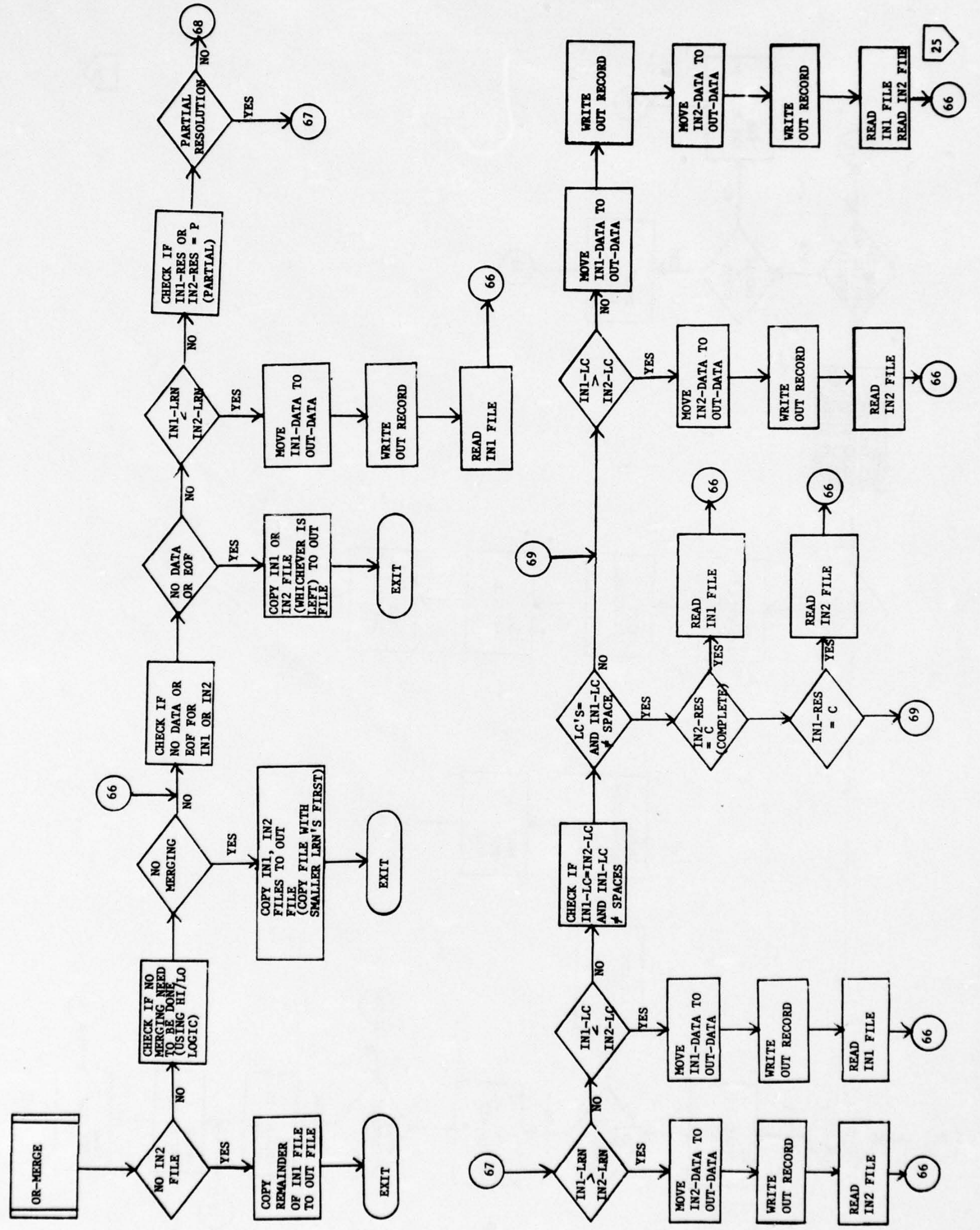

56
MOVE OUT TO SET FILE INDICATOR
OUT ≠ 0
NO
57
YES
MOVE 1 TO FILE-IND (OUT)
57
RESET CON-STRAINT FILE INDICATOR, OUT
CHECK IF CONSTRAINT FILE IS AN EFFICIENCY FILE
EFFICIENCY
NO
EXIT
YES
TURN ON SET EFFICIENCY INDICATOR
TURN OFF CONSTRAINT EFFICIENCY INDICATOR
EXIT
20

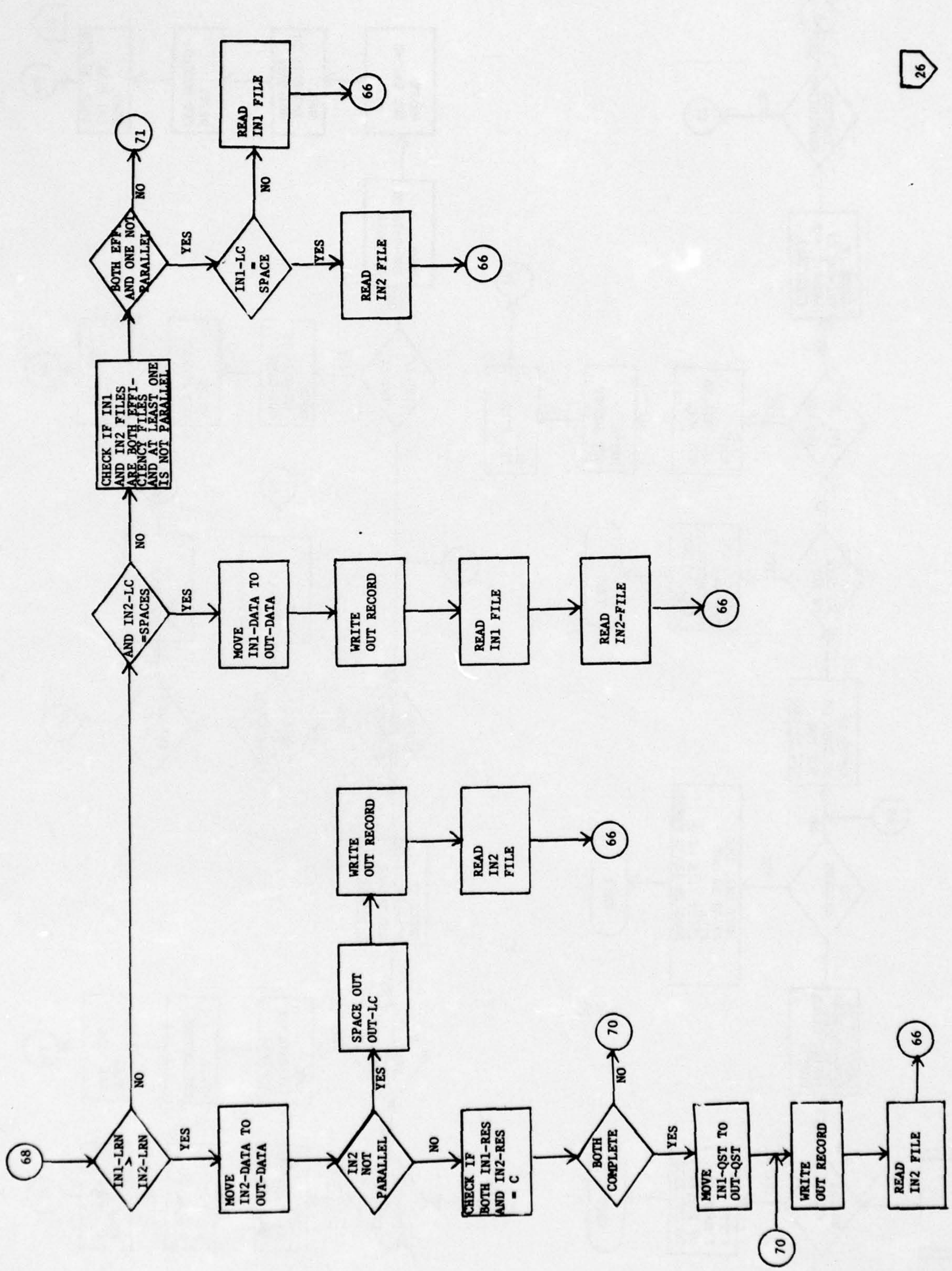

MRG-GROUPS
CHECK IF THERE IS A GROUP HIT FILE TO BE MERGED WITH SET HIT FILE
GROUP FILE
YES
NO
SET OUT = INDICATOR OF SET HIT FILE
59
SET IN1 = INDICATOR OF EXISTING GROUP FILE
SET IN2 = INDICATOR OF SET FILE
SET NUM. HITS INDICATOR = 0
CHECK IF GROUP FILE IS AN EFFICIENCY FILE
58
58
EFFICIENCY
YES
NO
CHECK IF SET FILE IS AN EFFICIENCY FILE
EFFICIENCY
NO
MERGING PRELIM.
OR-MERGE
END-MERGING
59
CHECK IF SET FILE IS AN EFFICIENCY FILE
SWITCH IN1 AND IN2 VALUES
RESET GROUP AND SET FILE EFFICIENCY INDICATORS
MERGING PRELIM.
NOT-MERGE
END-MERGING
59
EFFICIENCY
YES
NO
MERGING PRELIM.
NOT-MERGE
END-MERGING
59
MERGING PRELIM.
AND-MERGE
END-MERGING
59
21

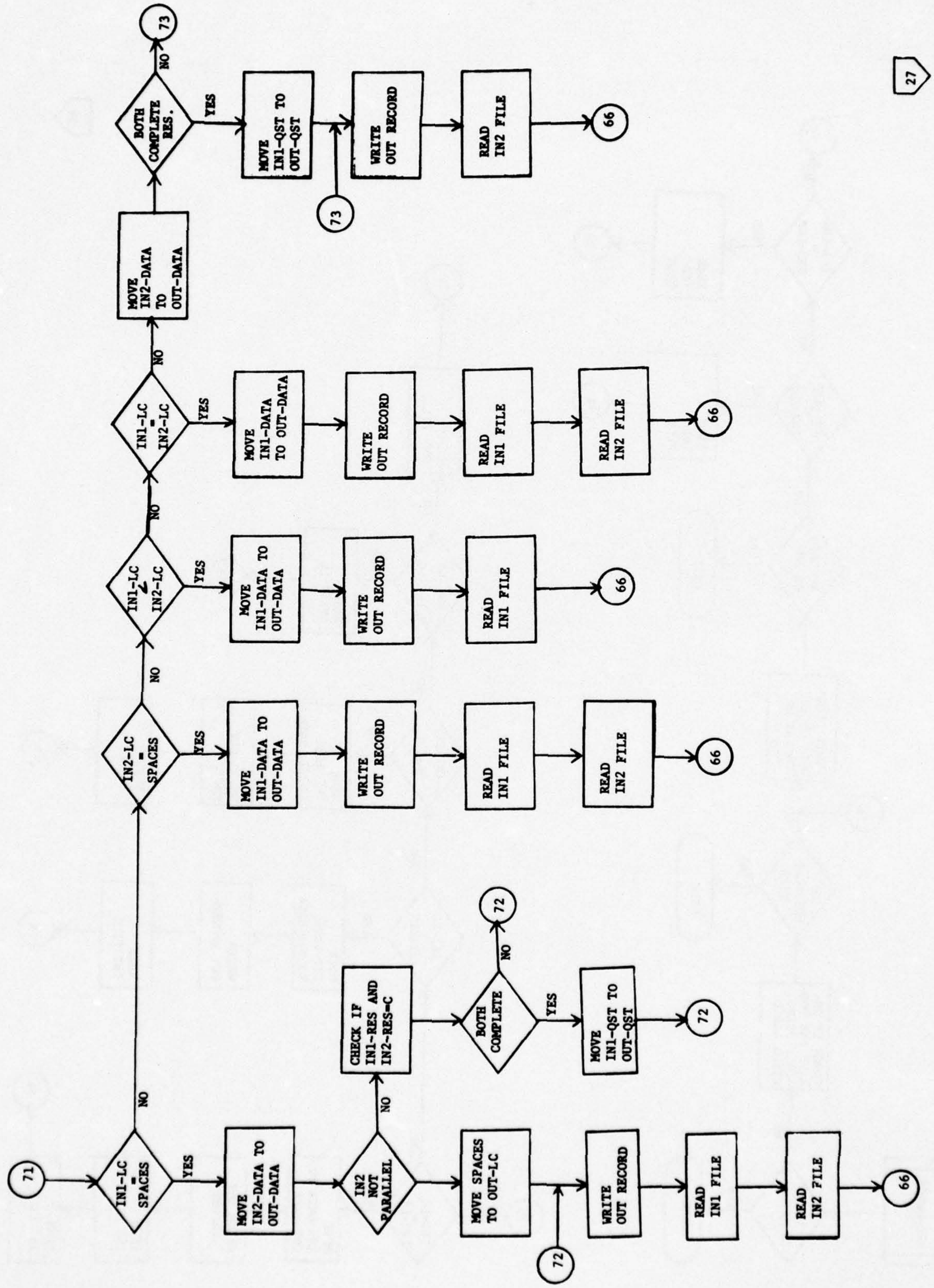

59
MOVE OUT TO GROUP FILE INDICATOR
OUT ≠ 0
YES
NO
60
MOVE HIGH-IND (OUT) TO GRP-HIGH
MOVE LOW-IND (OUT) TO GRP-LOW
MOVE 1 TO FILE-IND (OUT)
60
RESET SET FILE INDICATOR, OUT
CHECK IF SET FILE IS AN EFFI-CIENCY FILE
61
61
EFFICIENCY
YES
NO
EXIT
TURN ON GROUP FILE EFFICIENCY INDICATOR
TURN OFF SET FILE EFFICIENCY INDICATOR
EXIT
22

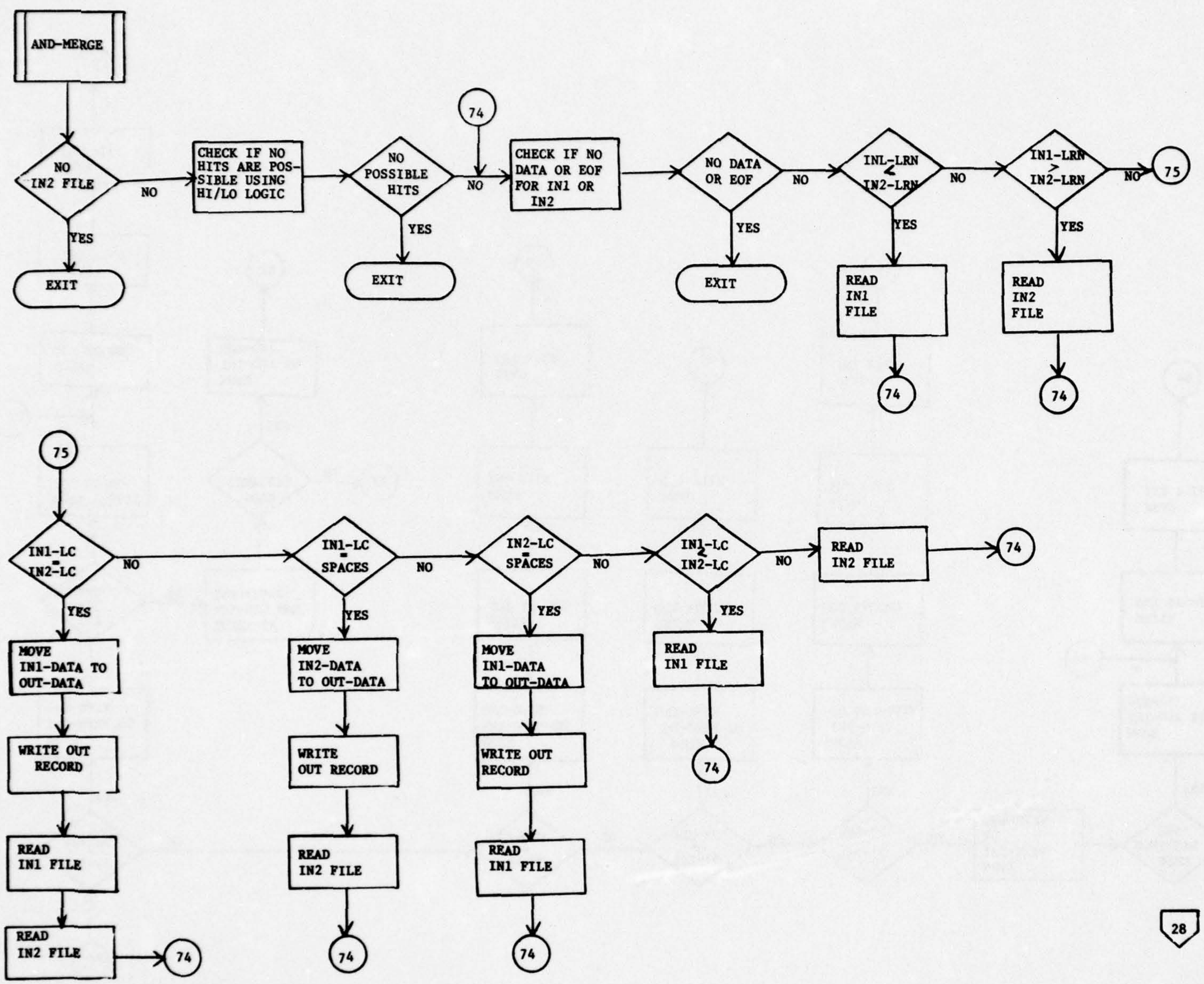

MRG-QUESTIONS
CHECK IF THERE IS A QUESTION HIT FILE TO BE MERGED WITH GROUP HIT FILE
QUESTION FILE
YES
NO
SET OUT = INDICATOR OF GROUP HIT FILE
63
SET IN1 = INDICATOR OF EXISTING QUESTION FILE
SET IN2 = INDICATOR OF GROUP FILE
SET IN2-RES = QUESTION RESOLUTION
MERGING PRELIM.
OR-MERGE
62
62
END-MERGING
63
MOVE OUT TO QST FILE INDICATOR
OUT ≠ 0
MOVE 1 TO FILE-IND (OUT)
64
64
RESET IN1-RES, IN2-RES QST FILE INDICATOR, OUT
CHECK IF NO MORE MERGING OR END OF CONSTRAINT FILE
NO MERGE OR EOF
YES
NO
WRITE-HIT-FILE
EXIT
EXIT
23


MERGING PRELIM.
OPEN IN1 FILE
SET IN1 SWITCH, INDICATORS
OPEN IN2 FILE
SET IN2 SWITCH, INDICATORS
OPEN OUT FILE
SET OUT SWITCH, INDICATORS
READ IN1 FILE
READ IN2 FILE
EXIT
END-MERGING
CHECK IF HIT MERGING STILL BEING DONE (HIT-INDIC=0)
HIT MERGING
YES
EXIT
NO
CLOSE SCRATCH FILES IN OPPOSITE ORDER OF OPENING
CHECK IF SOME HITS ON OUT FILE
SOME HITS
YES
MOVE OUT-REC-ID TO HIGH-IND(OUT)
65
NO
65
RESET FILE-IND FOR IN1, IN2
EXIT
24

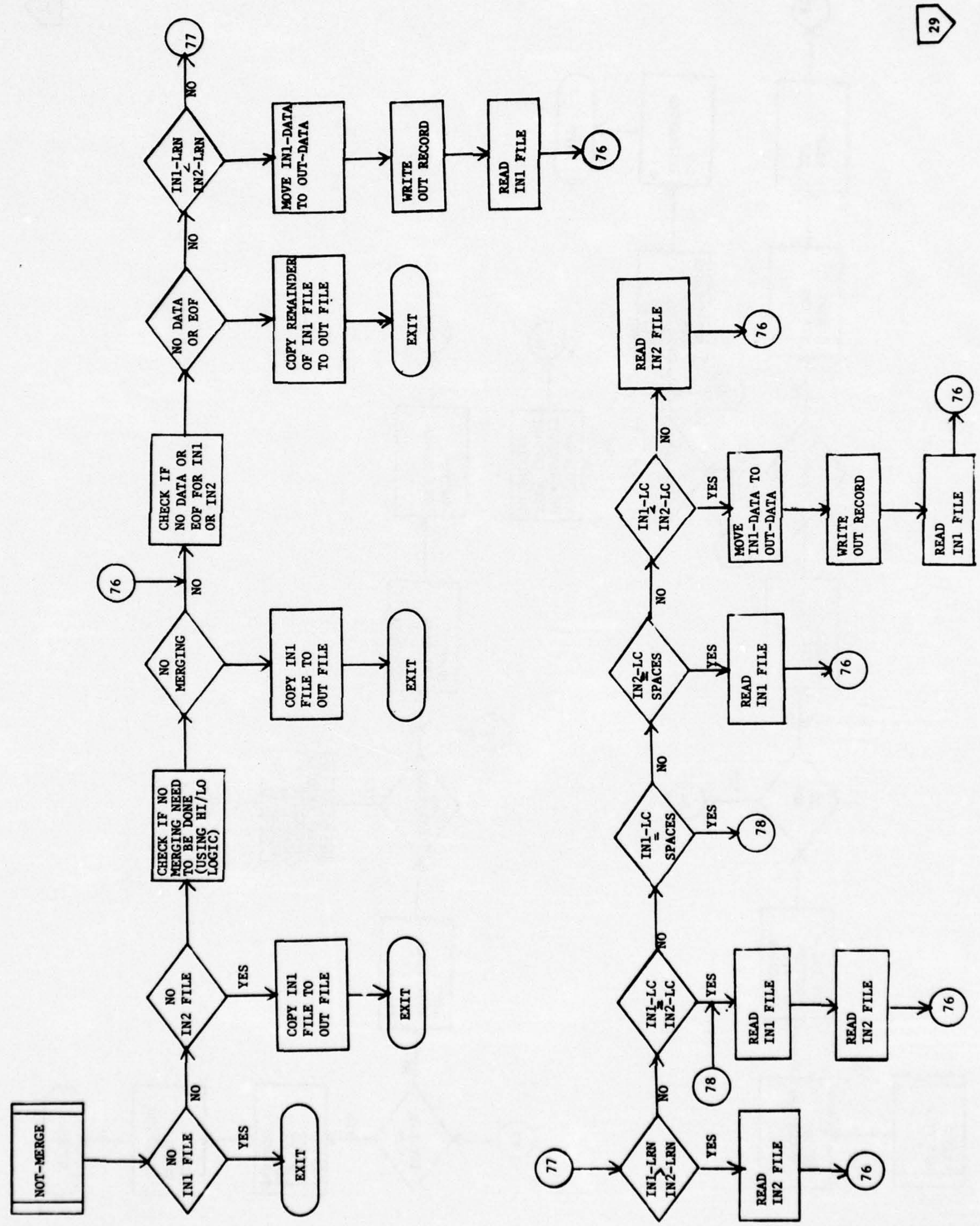

OR-MERGE
NO IN2 FILE
YES
COPY REMAINDER OF IN1 FILE TO OUT FILE
EXIT
NO
CHECK IF NO MERGING NEED TO BE DONE (USING HI/LO LOGIC)
NO MERGING
YES
COPY IN1, IN2 FILES TO OUT FILE (COPY FILE WITH SMALLER LRN'S FIRST)
EXIT
66
NO
CHECK IF NO DATA OR EOF FOR IN1 OR IN2
NO DATA OR EOF
YES
COPY IN1 OR IN2 FILE (WHICHEVER IS LEFT) TO OUT FILE
EXIT
NO
IN1-LRN < IN2-LRN
YES
MOVE IN1-DATA TO OUT-DATA
WRITE OUT RECORD
READ IN1 FILE
66
NO
CHECK IF IN1-RES OR IN2-RES = P (PARTIAL)
PARTIAL RESOLUTION
YES
67
NO
68
67
IN1-LRN > IN2-LRN
YES
MOVE IN2-DATA TO OUT-DATA
WRITE OUT RECORD
READ IN2 FILE
66
NO
IN1-LC < IN2-LC
YES
MOVE IN1-DATA TO OUT-DATA
WRITE OUT RECORD
READ IN1 FILE
66
NO
CHECK IF IN1-LC=IN2-LC AND IN1-LC ≠ SPACES
LC'S= AND IN1-LC ≠ SPACE
YES
IN2-RES = C (COMPLETE)
YES
READ IN1 FILE
66
IN1-RES = C
YES
READ IN2 FILE
66
69
NO
69
IN1-LC > IN2-LC
YES
MOVE IN2-DATA TO OUT-DATA
WRITE OUT RECORD
READ IN2 FILE
66
NO
MOVE IN1-DATA TO OUT-DATA
WRITE OUT RECORD
MOVE IN2-DATA TO OUT-DATA
WRITE OUT RECORD
READ IN1 FILE READ IN2 FILE
66
25

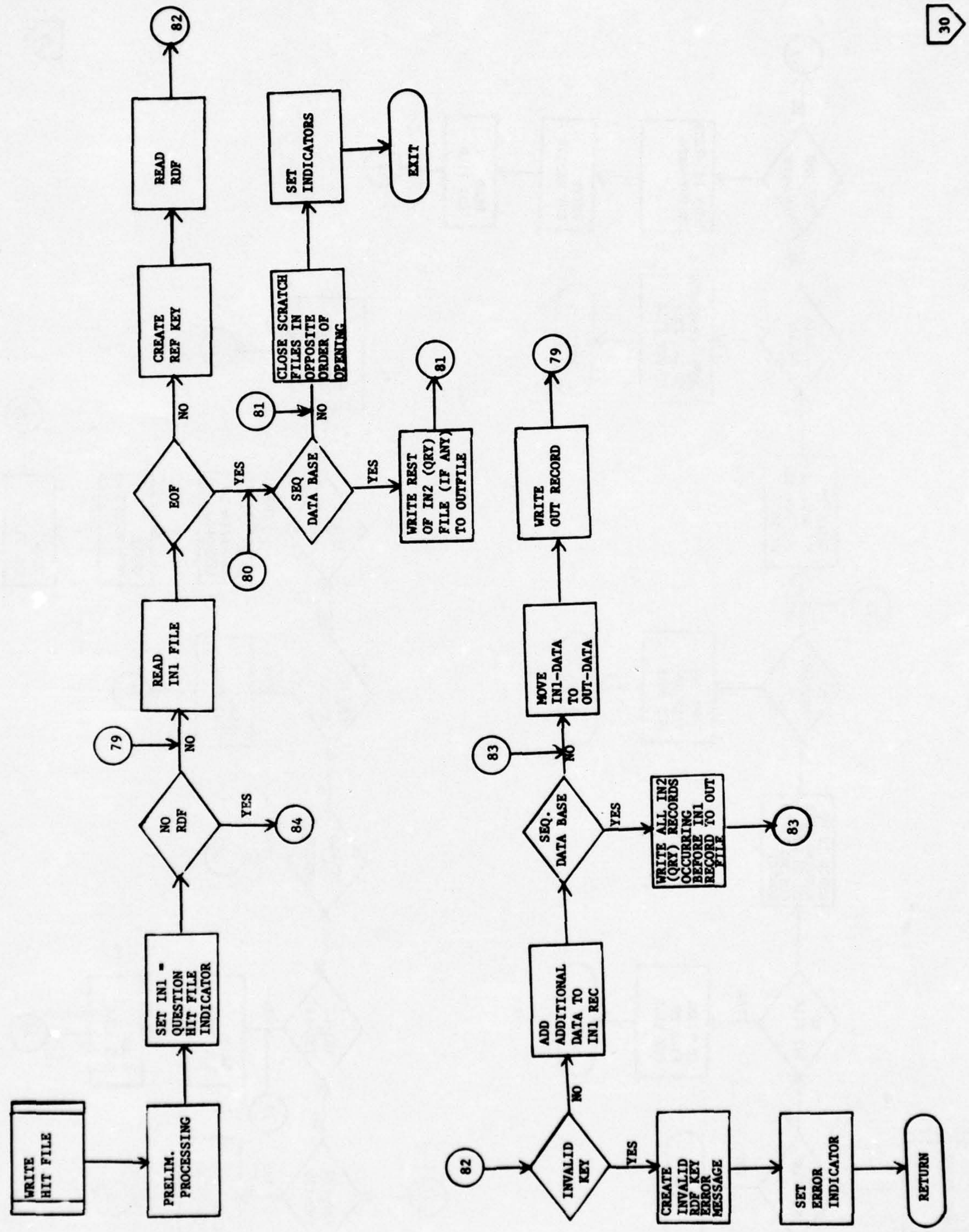

68
IN1-LRN > IN2-LRN
YES
NO
MOVE IN2-DATA TO OUT-DATA
IN2 NOT PARALLEL
YES
NO
SPACE OUT OUT-LC
WRITE OUT RECORD
READ IN2 FILE
66
CHECK IF BOTH IN1-RES AND IN2-RES = C
BOTH COMPLETE
NO
70
YES
MOVE IN1-QST TO OUT-QST
70
WRITE OUT RECORD
READ IN2 FILE
66
AND IN2-LC =SPACES
YES
NO
MOVE IN1-DATA TO OUT-DATA
WRITE OUT RECORD
READ IN1 FILE
READ IN2-FILE
66
CHECK IF IN1 AND IN2 FILES ARE BOTH EFFI-CIENCY FILES AND AT LEAST ONE IS NOT PARALLEL
BOTH EFF. AND ONE NOT PARALLEL
NO
71
YES
IN1-LC = SPACE
NO
READ IN1 FILE
66
YES
READ IN2 FILE
66
26

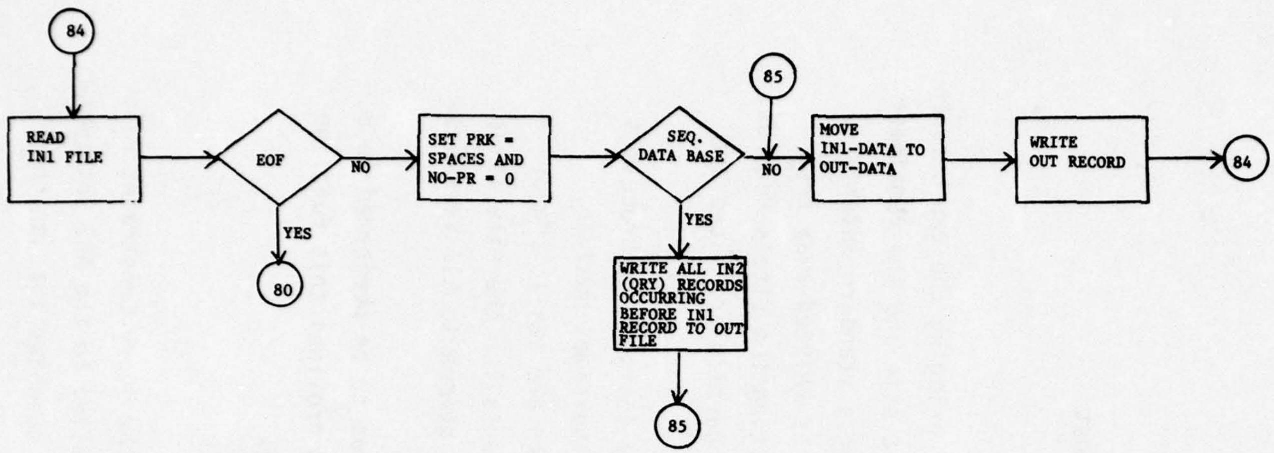

71
IN1-LC = SPACES
YES
NO
MOVE IN2-DATA TO OUT-DATA
IN2 NOT PARALLEL
NO
CHECK IF IN1-RES AND IN2-RES=C
BOTH COMPLETE
YES
NO
72
MOVE IN1-QST TO OUT-QST
72
MOVE SPACES TO OUT-LC
72
WRITE OUT RECORD
READ IN1 FILE
READ IN2 FILE
66
IN2-LC = SPACES
YES
NO
MOVE IN1-DATA TO OUT-DATA
WRITE OUT RECORD
READ IN1 FILE
READ IN2 FILE
66
IN1-LC < IN2-LC
YES
NO
MOVE IN1-DATA TO OUT-DATA
WRITE OUT RECORD
READ IN1 FILE
66
IN1-LC = IN2-LC
YES
NO
MOVE IN1-DATA TO OUT-DATA
WRITE OUT RECORD
READ IN1 FILE
READ IN2 FILE
66
MOVE IN2-DATA TO OUT-DATA
BOTH COMPLETE RES.
YES
NO
73
MOVE IN1-QST TO OUT-QST
73
WRITE OUT RECORD
READ IN2 FILE
66
27



AND-MERGE
NO IN2 FILE
YES
EXIT
NO
CHECK IF NO HITS ARE POSSIBLE USING HI/LO LOGIC
NO POSSIBLE HITS
YES
EXIT
74
NO
CHECK IF NO DATA OR EOF FOR IN1 OR IN2
NO DATA OR EOF
YES
EXIT
NO
INL-LRN < IN2-LRN
YES
READ IN1 FILE
74
NO
IN1-LRN > IN2-LRN
YES
READ IN2 FILE
74
NO
75
75
IN1-LC = IN2-LC
YES
MOVE IN1-DATA TO OUT-DATA
WRITE OUT RECORD
READ IN1 FILE
READ IN2 FILE
74
NO
IN1-LC = SPACES
YES
MOVE IN2-DATA TO OUT-DATA
WRITE OUT RECORD
READ IN2 FILE
74
NO
IN2-LC = SPACES
YES
MOVE IN1-DATA TO OUT-DATA
WRITE OUT RECORD
READ IN1 FILE
74
NO
IN1-LC < IN2-LC
YES
READ IN1 FILE
74
NO
READ IN2 FILE
74
28


NOT-MERGE
NO IN1 FILE
YES
EXIT
NO
NO IN2 FILE
YES
COPY IN1 FILE TO OUT FILE
EXIT
NO
CHECK IF NO MERGING NEED TO BE DONE (USING HI/LO LOGIC)
NO MERGING
COPY IN1 FILE TO OUT FILE
EXIT
NO
76
CHECK IF NO DATA OR EOF FOR IN1 OR IN2
NO DATA OR EOF
COPY REMAINDER OF IN1 FILE TO OUT FILE
EXIT
NO
IN1-LRN < IN2-LRN
NO
77
MOVE IN1-DATA TO OUT-DATA
WRITE OUT RECORD
READ IN1 FILE
76
77
IN1-LRN > IN2-LRN
YES
READ IN2 FILE
76
NO
IN1-LC = IN2-LC
YES
78
READ IN1 FILE
READ IN2 FILE
76
NO
IN1-LC = SPACES
YES
78
NO
IN2-LC = SPACES
YES
READ IN1 FILE
76
NO
IN1-LC < IN2-LC
YES
MOVE IN1-DATA TO OUT-DATA
WRITE OUT RECORD
READ IN1 FILE
76
NO
READ IN2 FILE
76
29


WRITE HIT FILE
PRELIM. PROCESSING
SET IN1 - QUESTION HIT FILE INDICATOR
NO RDF
YES
84
79
NO
READ IN1 FILE
80
EOF
YES
NO
CREATE REF KEY
READ RDF
82
SEQ DATA BASE
81
NO
YES
WRITE REST OF IN2 (QRY) FILE (IF ANY) TO OUTFILE
81
CLOSE SCRATCH FILES IN OPPOSITE ORDER OF OPENING
SET INDICATORS
EXIT
82
INVALID KEY
YES
CREATE INVALID RDF KEY ERROR MESSAGE
SET ERROR INDICATOR
RETURN
NO
ADD ADDITIONAL DATA TO IN1 REC
SEQ. DATA BASE
YES
WRITE ALL IN2 (QRY) RECORDS OCCURRING BEFORE IN1 RECORD TO OUT FILE
83
83
NO
MOVE IN1-DATA TO OUT-DATA
WRITE OUT RECORD
79
30


84
READ
IN1 FILE
EOF
YES
80
NO
SET PRK =
SPACES AND
NO-PR = 0
SEQ.
DATA BASE
YES
WRITE ALL IN2
(QRY) RECORDS
OCCURRING
BEFORE IN1
RECORD TO OUT
FILE
85
NO
85
MOVE
IN1-DATA TO
OUT-DATA
WRITE
OUT RECORD
84

31

A.7 TITLE: ADS INTERFACE MODULES

The following are all ADS Modules;

NLDLADS	-	ADS Module at DPSCLANT (never installed)
NLDPADS	-	ADS Module at DPSCPAC
ADSFACE	-	ADS Module at NMCSA
ADS370	-	Test Version of ADSFACE
ADS7045	-	Test Version of NLDPADS
ADSSHRM	-	ADSFACE with PCPs for Test

A.7.1 PURPOSE

The ADS Modules perform the function of bridging the gap between the uniquely formatted data files resident at a site and the remainder of the generalized NAVLIS system. By providing a standard interface, the burden of coping with non-standardization is shifted away from other NAVLIS modules. In addition to reformatting the site's records, conversions of data fields are performed. In particular, dates are converted from Julian to calendar formats.

In a sense, each ADS module is uniquely designed to process a specific file. Much of the program logic is the same, however.

In fact, differences between the modules are small. Major differences occur only in the access requirements for the site files. Processing of the constraint and hit files is common to all versions, as would be the SHRM Implementation.

Originally, secondary hit resolution was to be performed in a separate module, SHRM. A PCP to be described provides this function as an enhancement directly to the ADS modules.

A.7.2 INPUT

To define the query being processed, the Query Constraint File created by FSM, or if applicable SRM, is supplied to the ADS Interface Module. Hit resolution performed by PHRM is supplied through the Hit File. This file indicates which records are hits if complete

resolution is performed by PHRM. Otherwise the records are candidates with final resolution to be performed in the ADS Interface Module.

A.7.3 OUTPUT

Output of an ADS Interface Module consists of a series of Response Records passed one at a time back to ECM/RCM. Each record contains the LIRC and data value of a field required either in printing a report or performing a computation.

In addition, a count of hits by question is supplied to ECM/RCM as well as Return and Error Indicators.

A.7.4 RECORD FORMATS - Response Records returned to ECM/RCM have the following format (See Figure A7)

Bytes	Data Field
1-3	Query ID
4-6	Question Number
8	Asterisk Flag for non-printed items
9	Type Code: C,D, or J
10-11	Set Number
13-16	LIRC
17-18	Length of value in characters
19-114	LIRC Value

Hit File Records have the format - (See Figure A13)

Bytes	Data Field
1-3	Query ID
4-6	Question Number
7-8	Search Type Code
9	Constraint Type Code (L-Z)
10-11	Set Number
15	Whole/Prefix Search Switch
16-19	LIRC
27	Parallel Indicator, P if parallel

28	Efficiency Search Indicator, Space for not efficiency search
29-78	Search Value

C-Type Records (See Figure A2)

Byte	Data Field
1-3	Query ID
4-6	Question Number
8	Asterisk Flag
9	Always "C"
10-11	Set Number
13-16	First LIRC
33-36	Second LIRC

D-Type Records (See Figure A3)

Byte	Data Field
1-3	Query ID
4-6	Question Number
8	Asterisk Flag
9	Always "D"
10-11	Set Number
13-16	LIRC

J-Type Records (See Figure A4)

Byte	Data Field
1-3	Query ID
4-6	Question Number
8	Asterisk Flag
9	Always "J"
10-11	Set Number
13-16	LIRC
27	Parallel Indicator, P for Parallel LIRCs.

A-Type Records (See Figure A1)

Bytes	Data Field
1-3	Query ID
4-6	Question - Number
9	Always "A"
10-11	01 for first question of a merge 00 for last question of a merge

Format of database records is determined by the contents of LIRC-FINDER-TABLE. All LIRCs are identified with a starting location, length, line count, and number of occurrences.

A.7.5 STORAGE REQUIREMENTS

Program (less library modules) occupies 42,000 bytes on UNIVAC Series 70/45.

A.7.6 PROCESSING DESCRIPTION (Refer to ADSFACE Flowchart, page 145)

Processing within ADSFACE is conceptually divided into three main areas. Initial processing is devoted to reading in the constraints for a question along with the C,D, and J records indicating the output LIRCs needed. After the constraint and J-Record tables are built, the Hit File is read and records are retrieved from the data base. Fields are then extracted from the database record as indicated by entries in the J-Table. Response records are created and passed one at a time to ECM/RCM.

Building of constraint and J-Record tables occurs one question at a time. In the case of merged questions, all questions being merged are treated as a unit. The constraint table is built up by storing each input record of the question including A, C, D and L-Z type records in a table containing the input in card image format. When SHRM processing is included in the ADS Module, this procedure is somewhat more elaborate as will be described with the relevant PCP.

The J-Table is built up from C,D, and J type records. J type records are entered as they occur. Processing of C and D type records is delayed. After all constraint file records for the question have been made, a scan of the J-table is made. A test for LIRC indicator of "ALL" is made. If it is found, the LIRC finder table is used to enter J records for all LIRCs occurring in the database.

After the initial scan is made, a scan of the constraint table is made to pick up C and D type records. Before adding an entry to the J-table, a test is made to make sure that a LIRC is not duplicated. If the LIRC already is in the J-table and it appears on a D-type record, the sort key of the J entry is adjusted so that the entry will sort to the beginning of the table and cause that LIRC to be output before all LIRCs not occurring in D records.

When the constraint table scan is complete, the J-table is sorted into the sequence required for generating response records. At this point table building is complete.

When complete hit resolution has been performed by PHRM, processing of the Hit File is less complex. Processing required when SHRM capabilities are included will be described with the PCP.

Each hit record is read and a test for a new question is made. If a new question is encountered, control returns to the table building logic. Otherwise, the record is read from the database as indicated by the key and record number in the hit record. Some preliminary processing of the database record may be performed at this point. Both NLDPADS and NLDLADS process SHARP formatted databases. The LIRC finder table supplied is skeletal since locations of the fields are not fixed. A scan of the record is made by the routine SITUATE-SHARP-RECORD to fill in the locations of all fields in the record.

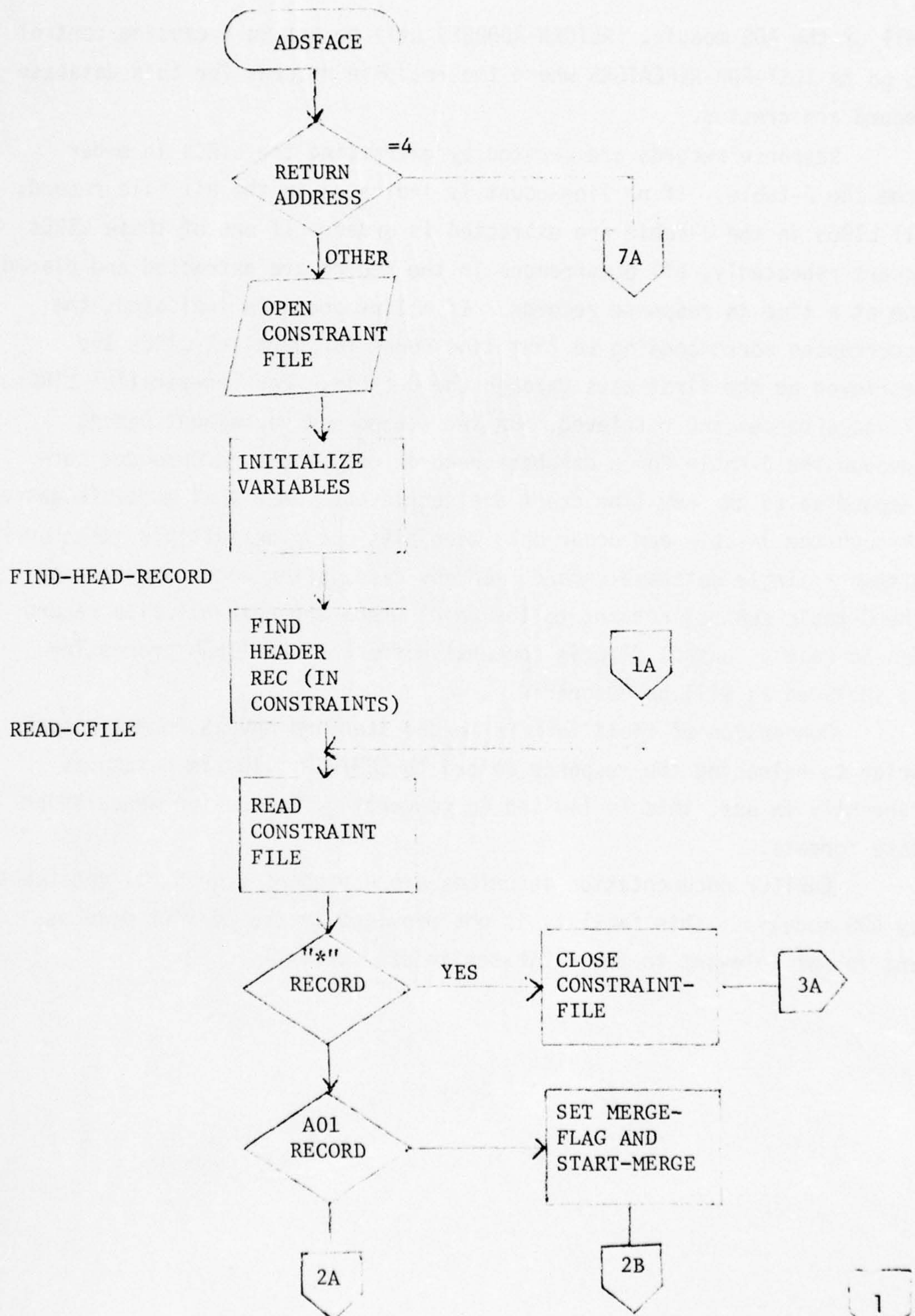
In the complete hit resolution case, response records can now be created. The first response record is always for LIRC 0000, the record key. This key is taken from the hit file record since the hashed key is not stored in the database. The variable RETURN-ADDRESS is used to indicate where to go after return to ECM and its subsequent

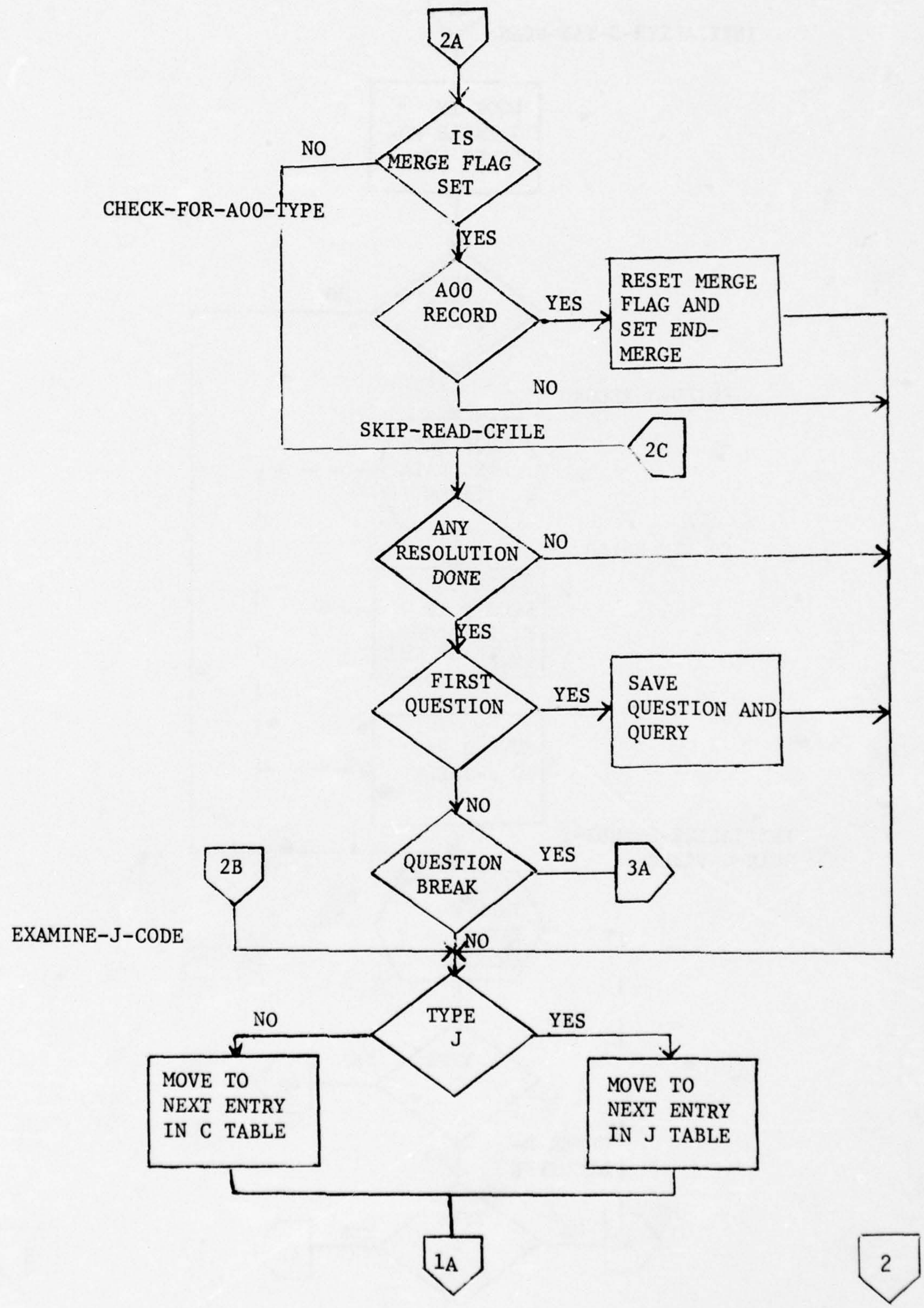
call of the ADS module. RETURN-ADDRESS will be set to 4 causing control to go to TEST-FOR-REPEATORS where the response records for this database record are created.

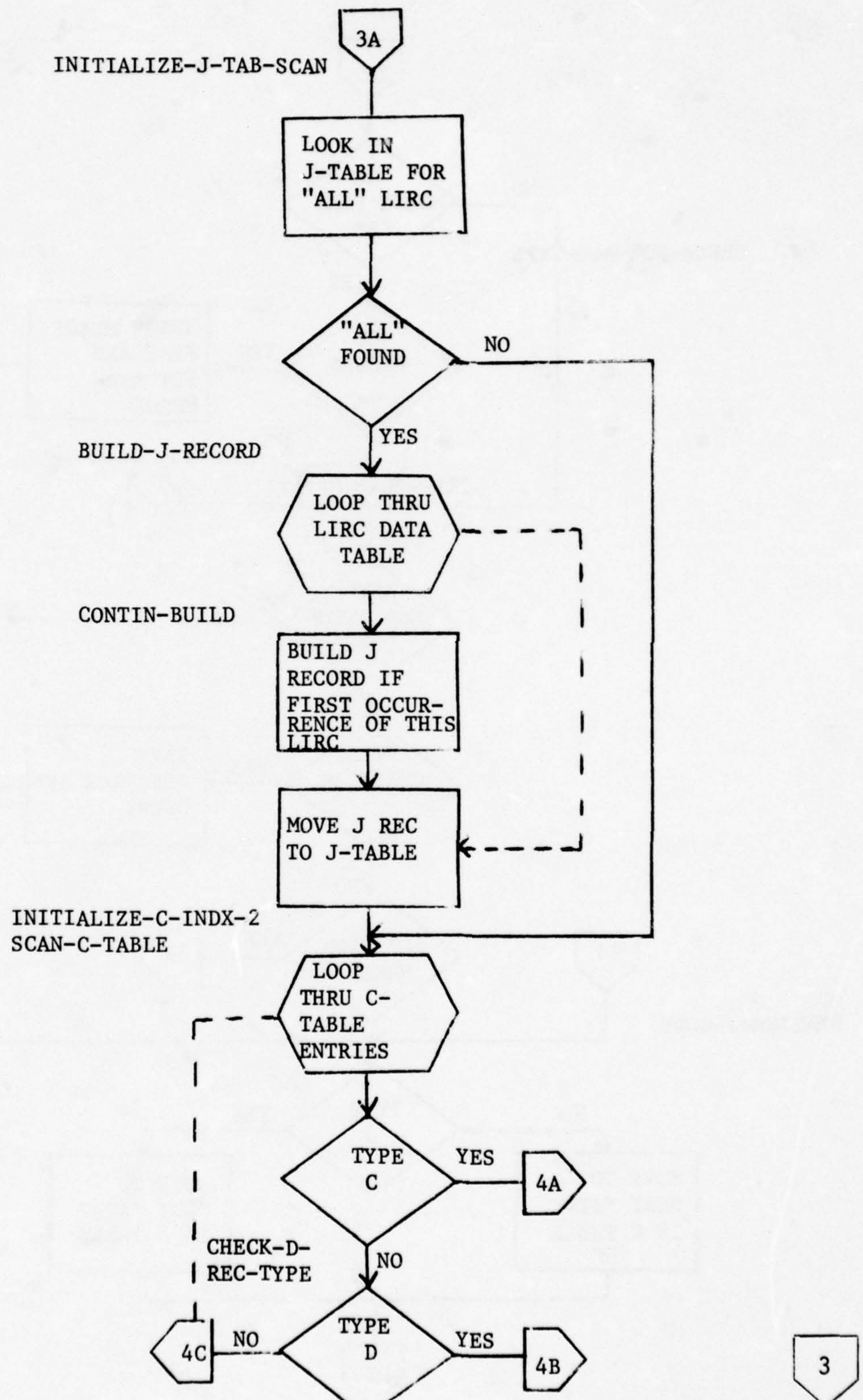
Response records are created by extracting the LIRCs in order from the J-table. If no line-count is indicated in the hit file record, all LIRCs in the J-table are extracted in order. If one of these LIRCs occurs repeatedly, all occurrences in the record are extracted and placed one at a time in response records. If a line count is indicated, the occurrences corresponding to that line count for parallel LIRCs are retrieved on the first pass through the J-table. For non-parallel LIRCs, all occurrences are retrieved. On the second and subsequent passes through the J-table for a database record, only those occurrences corresponding to the new line count are retrieved. Note that multiple passes through the J-table can occur only when hits occur on multiple line counts within a single database record. In any case, after each pass through the J-table control returns to the point where the next hit file record can be read. Control flow is somewhat different when SHRM processing is included as will be described.

Conversion of field formats to the standard NAVLIS format occurs prior to releasing the response record to ECM/RCM. In the databases currently in use, this is limited to converting from Julian to calendar date formats.

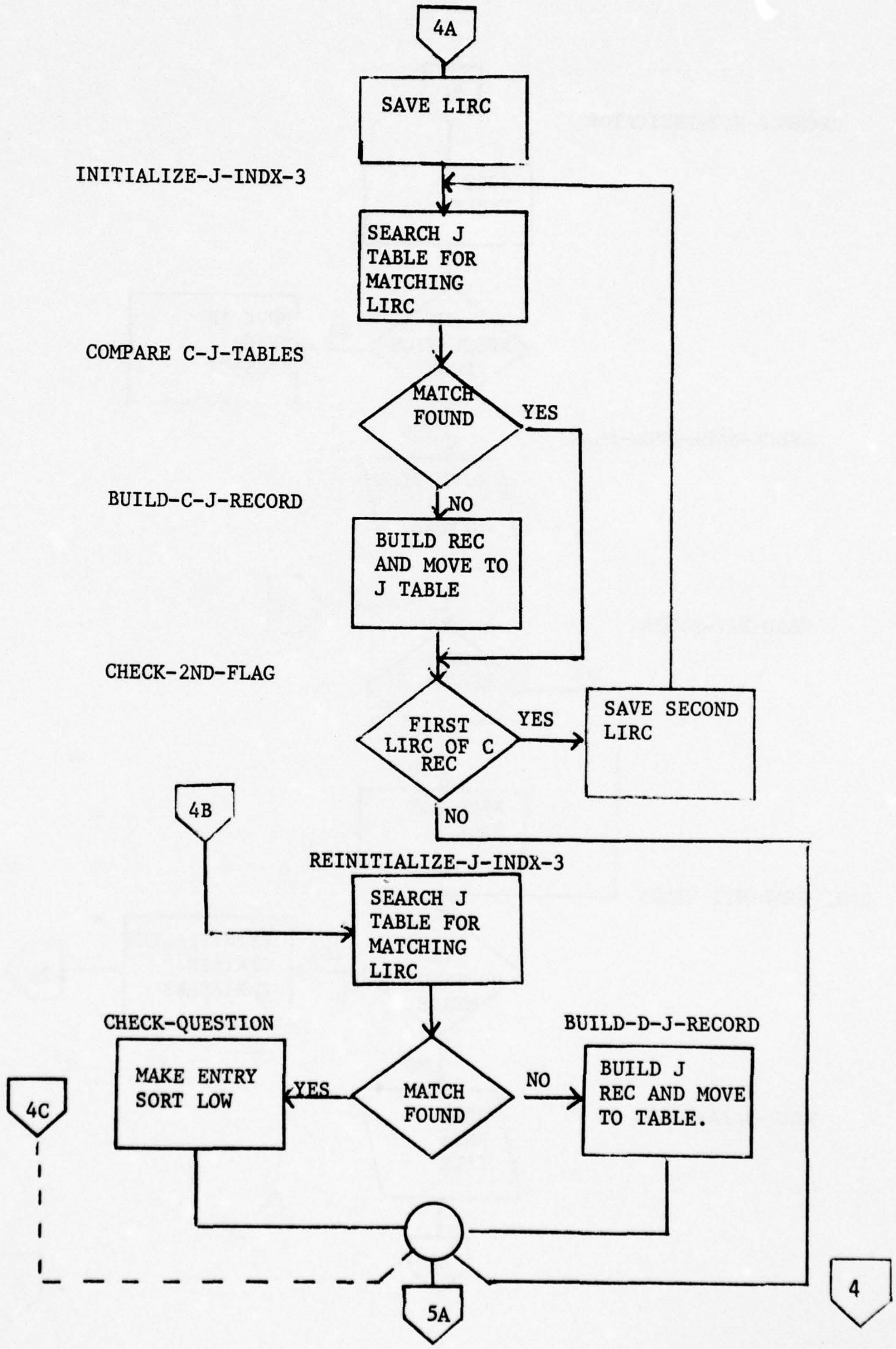
Earlier documentation describes processing of sequential databases by ADS modules. This facility is not provided in the current modules and is not relevant to the databases in use currently.

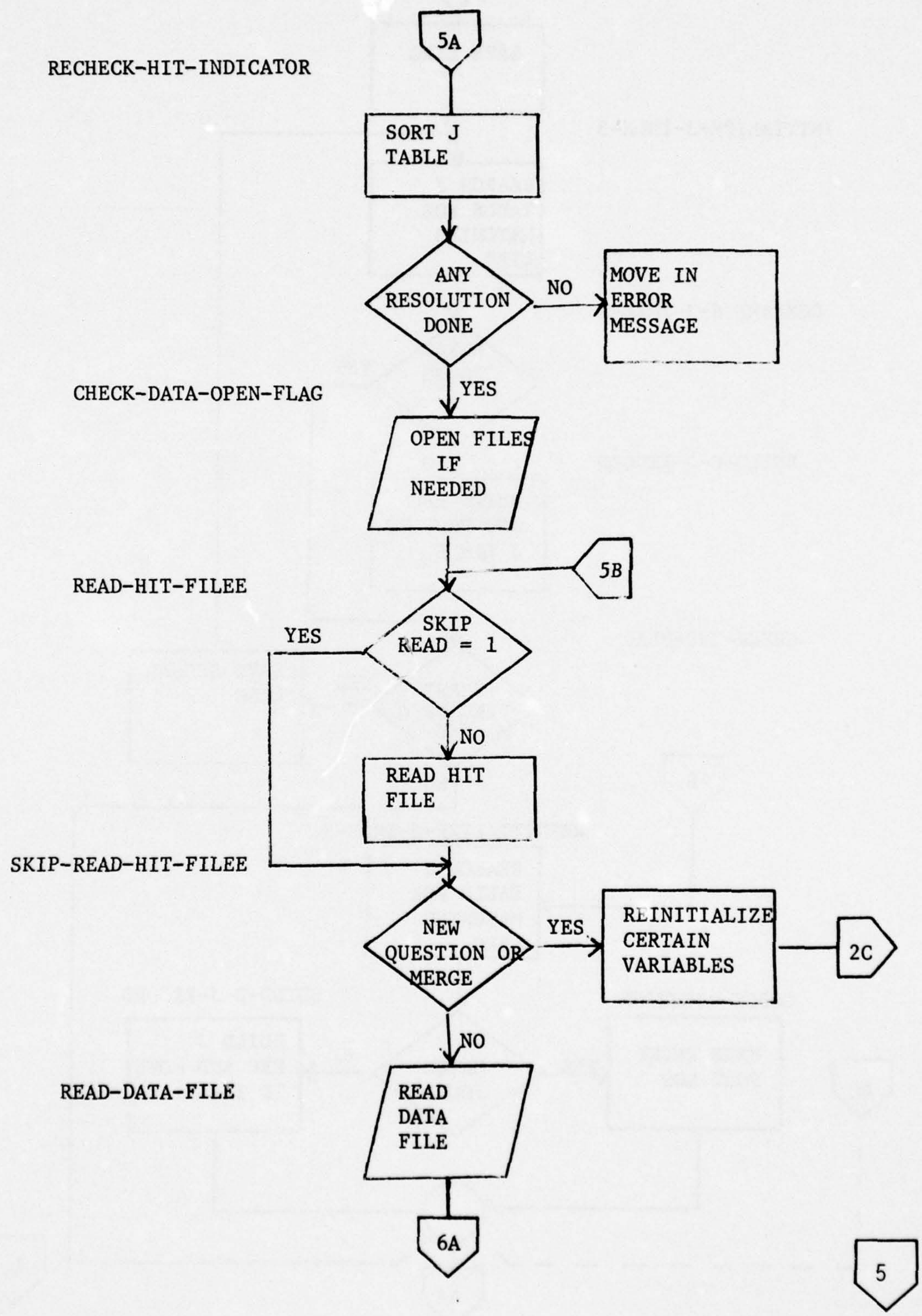

ADSFACE
RETURN ADDRESS
=4
7A
OTHER
OPEN CONSTRAINT FILE
INITIALIZE VARIABLES
FIND-HEAD-RECORD
FIND HEADER REC (IN CONSTRAINTS)
1A
READ-CFILE
READ CONSTRAINT FILE
"*" RECORD
YES
CLOSE CONSTRAINT-FILE
3A
A01 RECORD
SET MERGE-FLAG AND START-MERGE
2A
2B
1

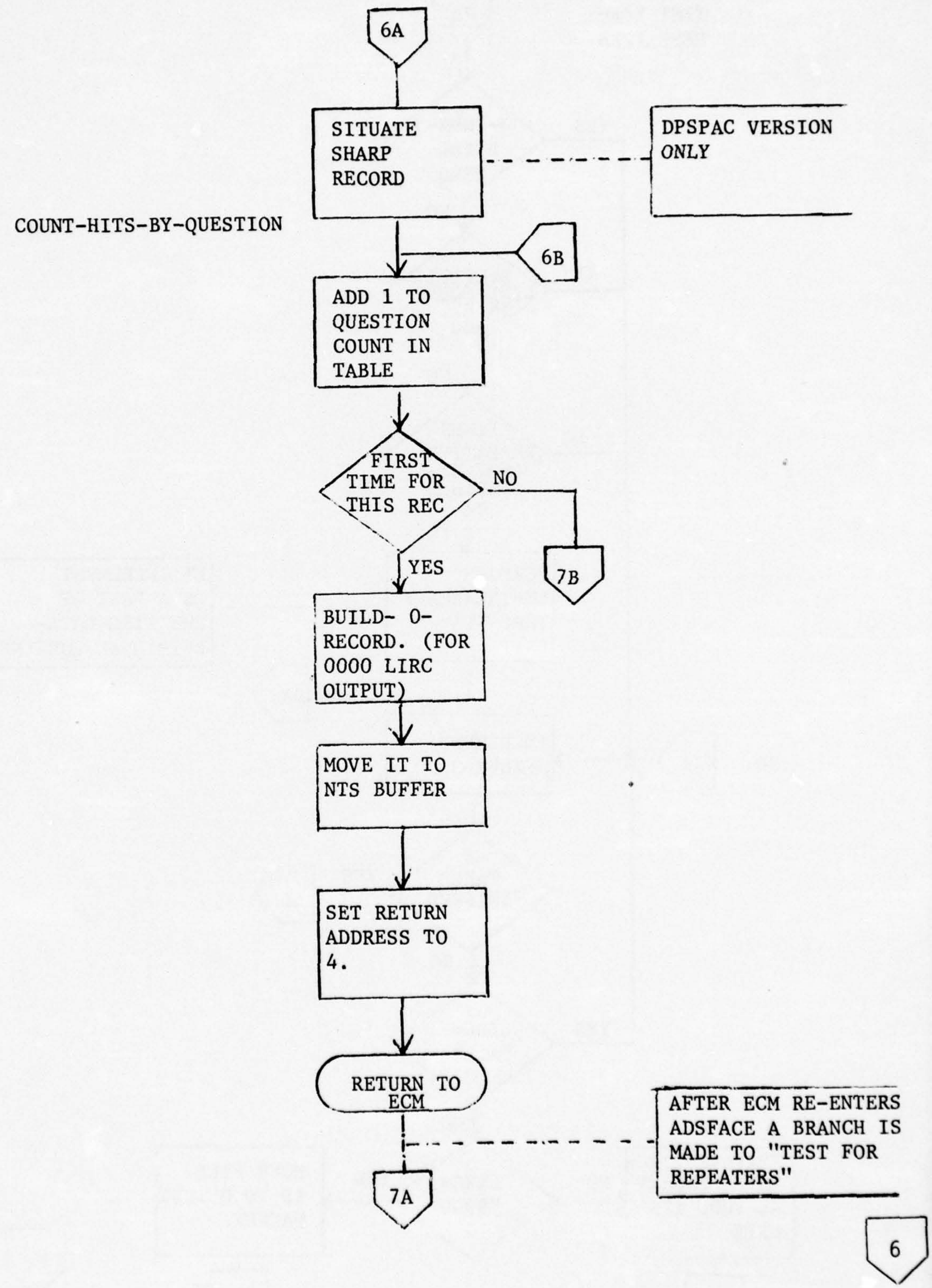

2A
IS MERGE FLAG SET
NO
YES
CHECK-FOR-A00-TYPE
A00 RECORD
YES
RESET MERGE FLAG AND SET END-MERGE
NO
SKIP-READ-CFILE
2C
ANY RESOLUTION DONE
NO
YES
FIRST QUESTION
YES
SAVE QUESTION AND QUERY
NO
QUESTION BREAK
YES
3A
2B
EXAMINE-J-CODE
NO
TYPE J
NO
YES
MOVE TO NEXT ENTRY IN C TABLE
MOVE TO NEXT ENTRY IN J TABLE
1A
2

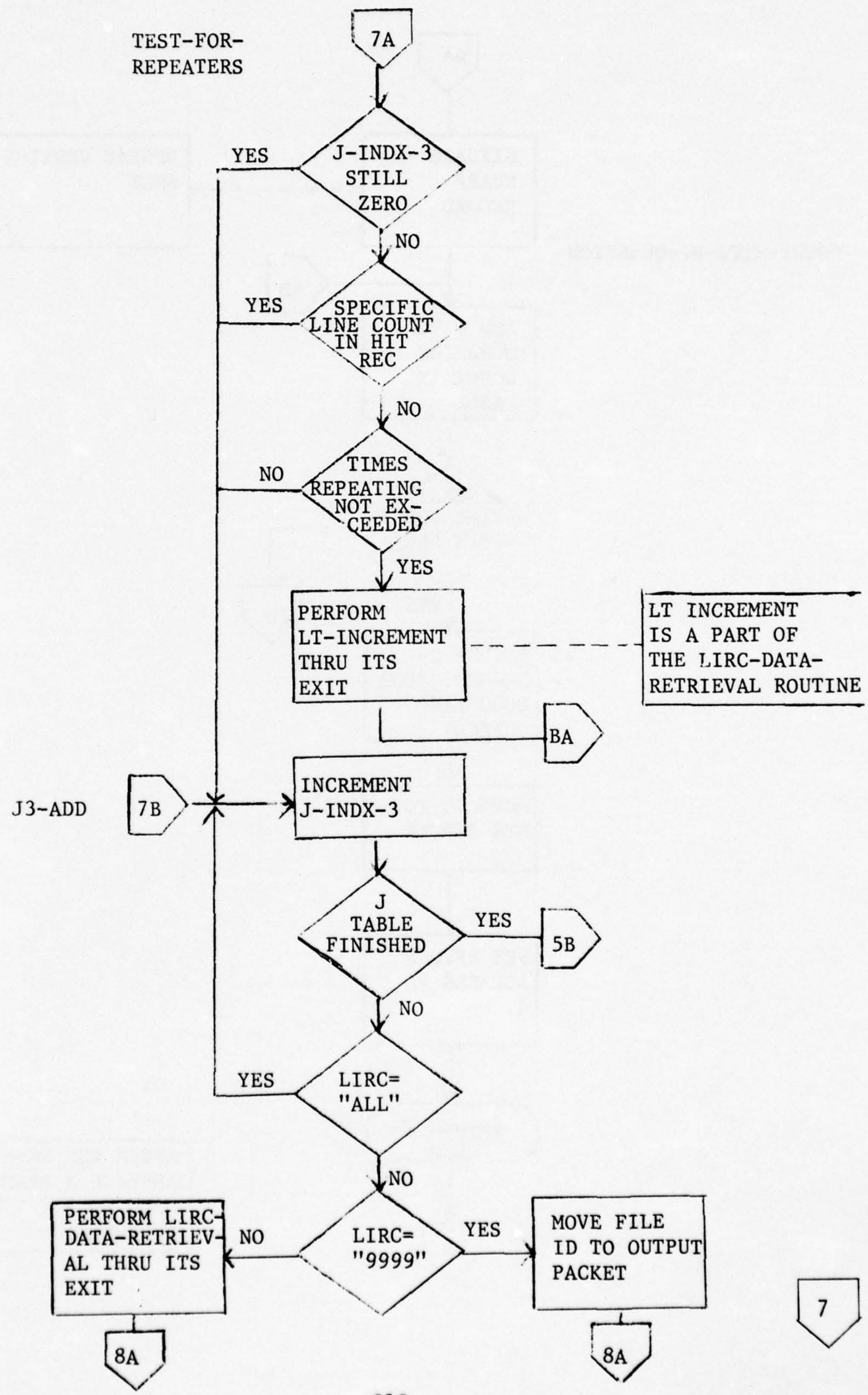


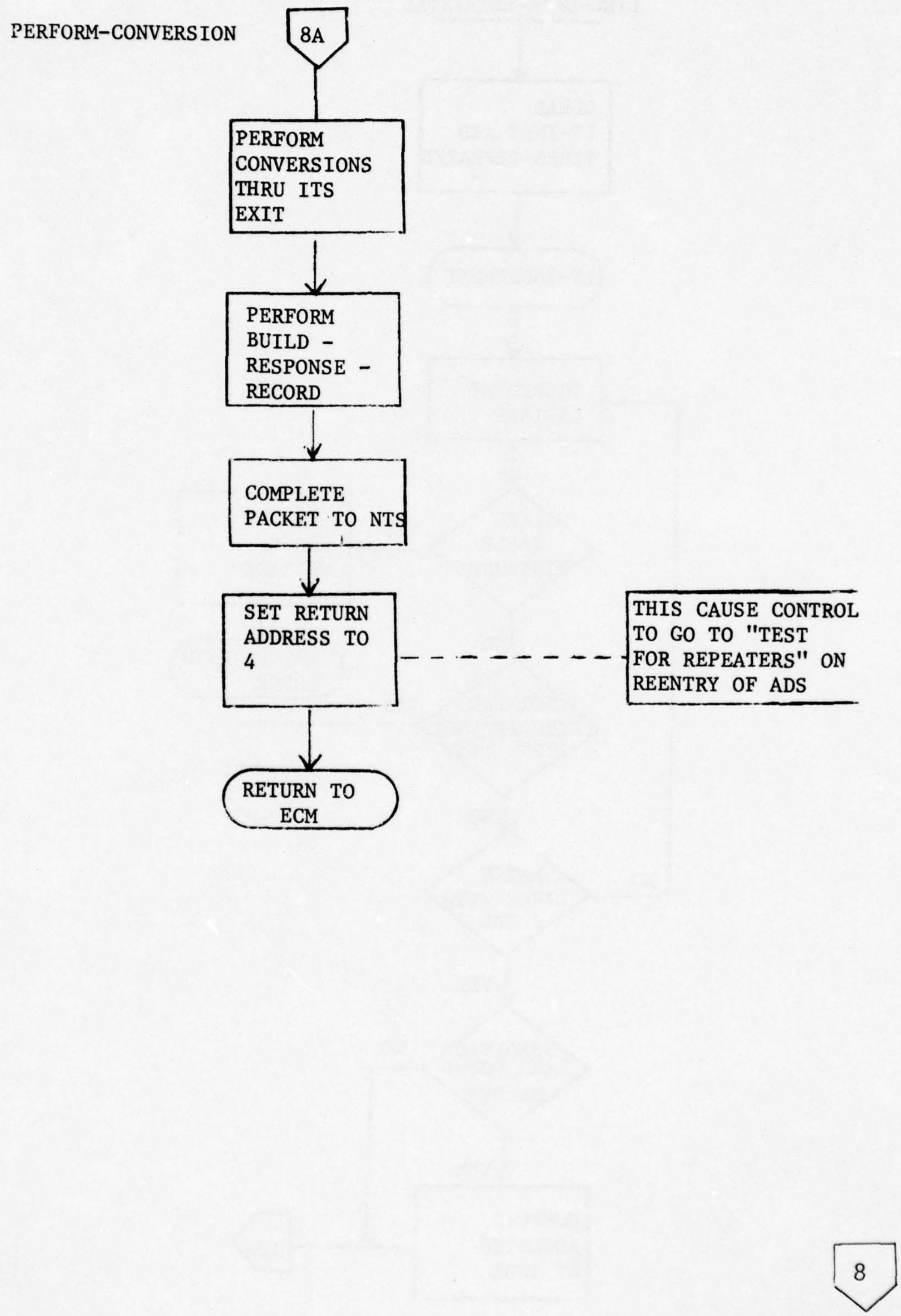
3A
INITIALIZE-J-TAB-SCAN
LOOK IN J-TABLE FOR "ALL" LIRC
"ALL" FOUND
NO
YES
BUILD-J-RECORD
LOOP THRU LIRC DATA TABLE
CONTIN-BUILD
BUILD J RECORD IF FIRST OCCURRENCE OF THIS LIRC
MOVE J REC TO J-TABLE
INITIALIZE-C-INDX-2
SCAN-C-TABLE
LOOP THRU C-TABLE ENTRIES
TYPE C
YES
4A
NO
CHECK-D-REC-TYPE
TYPE D
NO
4C
YES
4B
3

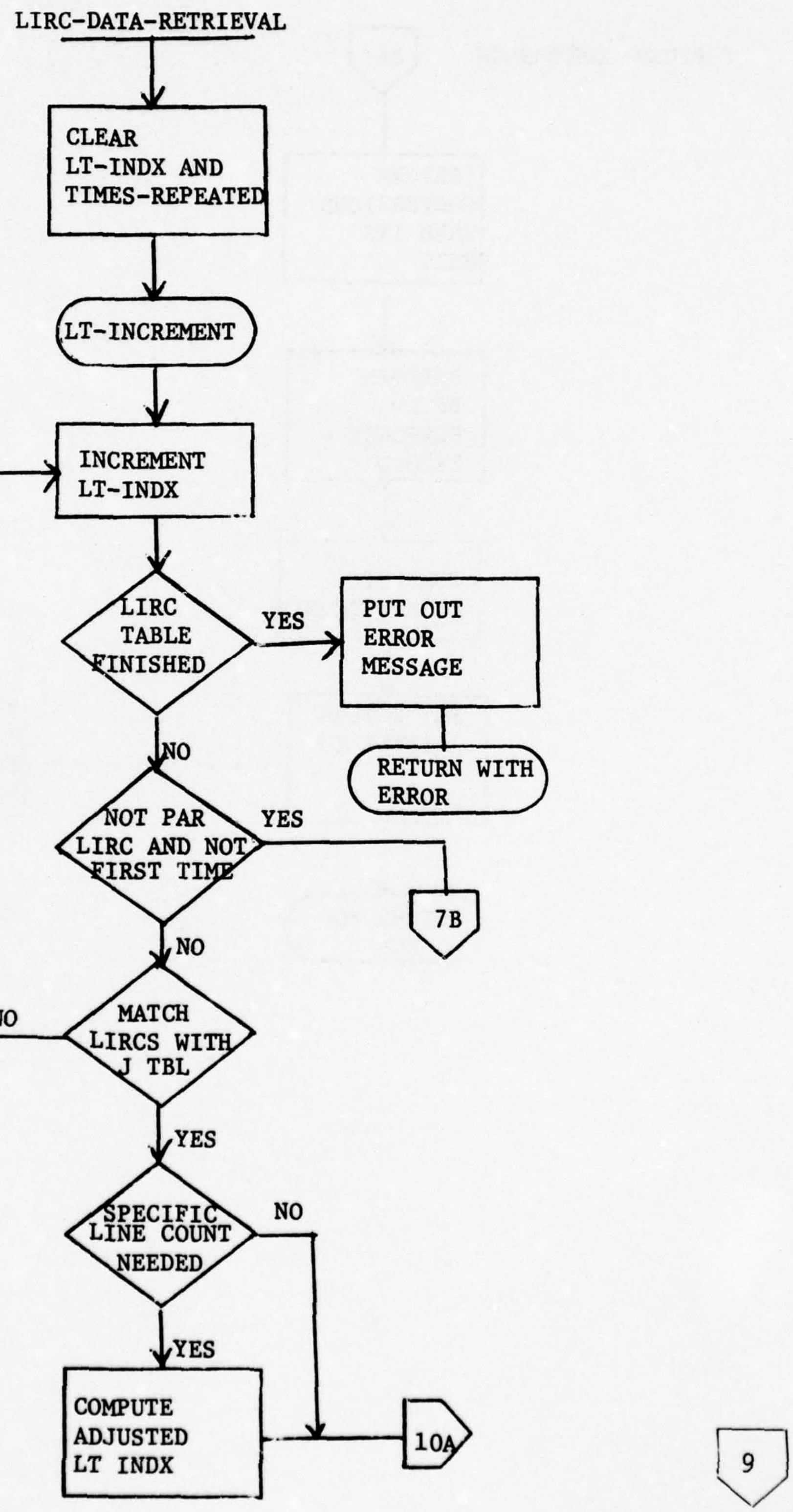

4A
SAVE LIRC
INITIALIZE-J-INDX-3
SEARCH J TABLE FOR MATCHING LIRC
COMPARE C-J-TABLES
MATCH FOUND
YES
NO
BUILD-C-J-RECORD
BUILD REC AND MOVE TO J TABLE
CHECK-2ND-FLAG
FIRST LIRC OF C REC
YES
SAVE SECOND LIRC
NO
4B
REINITIALIZE-J-INDX-3
SEARCH J TABLE FOR MATCHING LIRC
CHECK-QUESTION
MAKE ENTRY SORT LOW
YES
MATCH FOUND
NO
BUILD-D-J-RECORD
BUILD J REC AND MOVE TO TABLE.
4C
5A
4

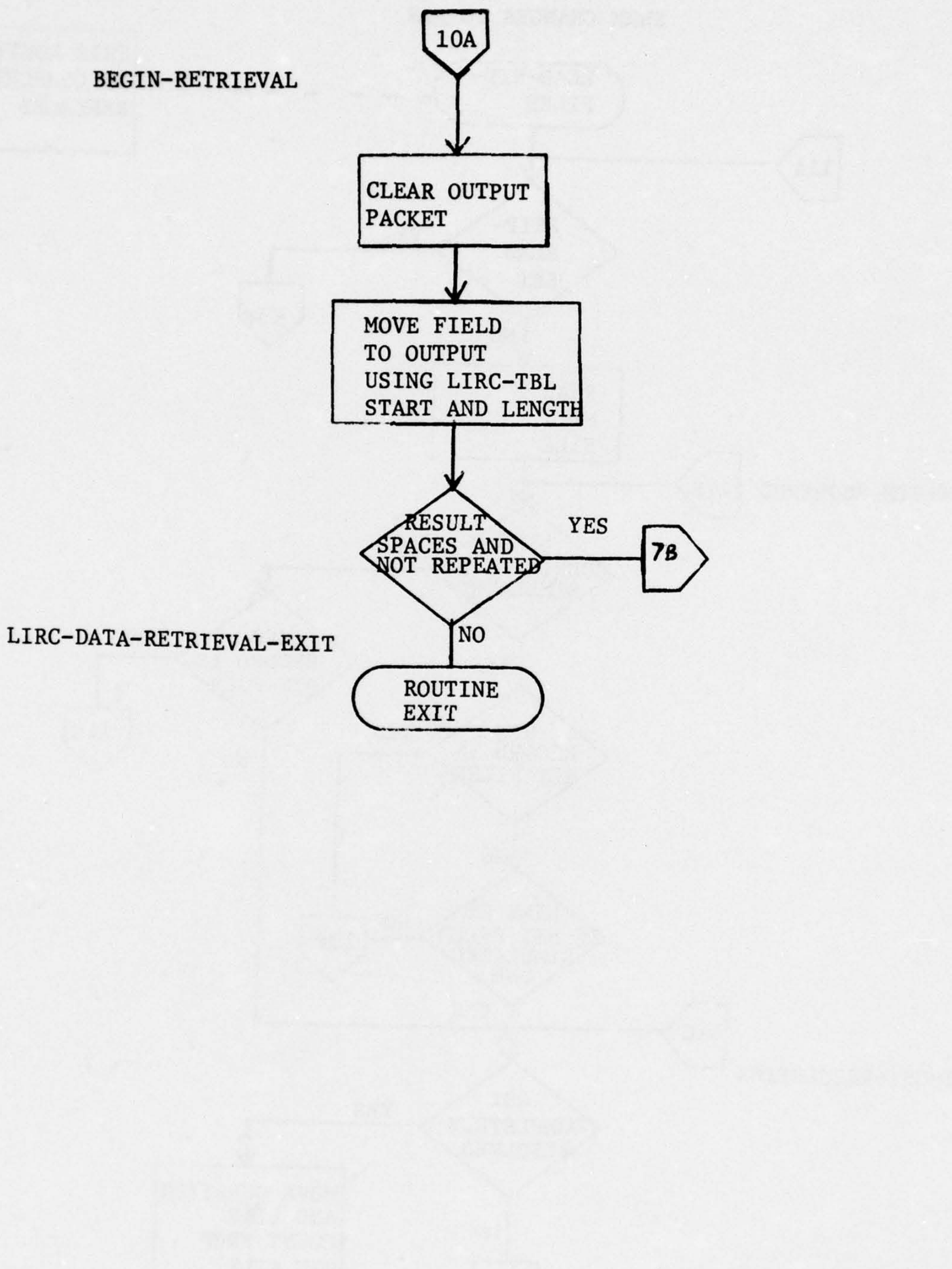

5A
RECHECK-HIT-INDICATOR
SORT J TABLE
ANY RESOLUTION DONE
NO
MOVE IN ERROR MESSAGE
YES
CHECK-DATA-OPEN-FLAG
OPEN FILES IF NEEDED
5B
READ-HIT-FILEE
SKIP READ = 1
YES
NO
READ HIT FILE
SKIP-READ-HIT-FILEE
NEW QUESTION OR MERGE
YES
REINITIALIZE CERTAIN VARIABLES
2C
NO
READ-DATA-FILE
READ DATA FILE
6A
5


6A
SITUATE SHARP RECORD
DPSPAC VERSION ONLY
COUNT-HITS-BY-QUESTION
6B
ADD 1 TO QUESTION COUNT IN TABLE
FIRST TIME FOR THIS REC
NO
7B
YES
BUILD- 0- RECORD. (FOR 0000 LIRC OUTPUT)
MOVE IT TO NTS BUFFER
SET RETURN ADDRESS TO 4.
RETURN TO ECM
AFTER ECM RE-ENTERS ADSFACE A BRANCH IS MADE TO "TEST FOR REPEATERS"
7A
6

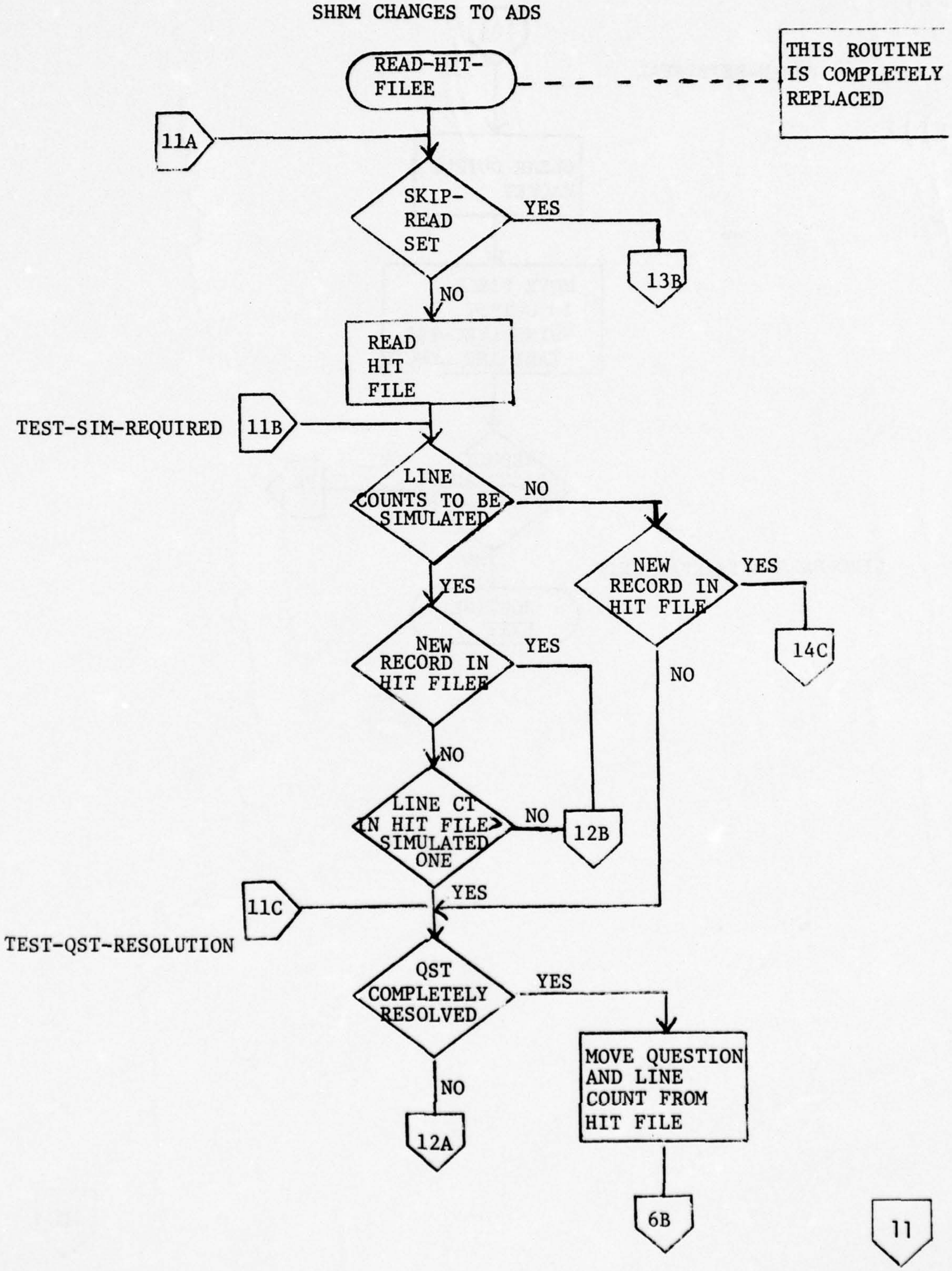

TEST-FOR-
REPEATERS
7A
J-INDX-3 STILL ZERO
YES
NO
SPECIFIC LINE COUNT IN HIT REC
YES
NO
TIMES REPEATING NOT EX-CEEDED
NO
YES
PERFORM LT-INCREMENT THRU ITS EXIT
LT INCREMENT IS A PART OF THE LIRC-DATA-RETRIEVAL ROUTINE
8A
J3-ADD
7B
INCREMENT J-INDX-3
J TABLE FINISHED
YES
5B
NO
LIRC= "ALL"
YES
NO
LIRC= "9999"
NO
YES
PERFORM LIRC-DATA-RETRIEV-AL THRU ITS EXIT
MOVE FILE ID TO OUTPUT PACKET
8A
8A
7

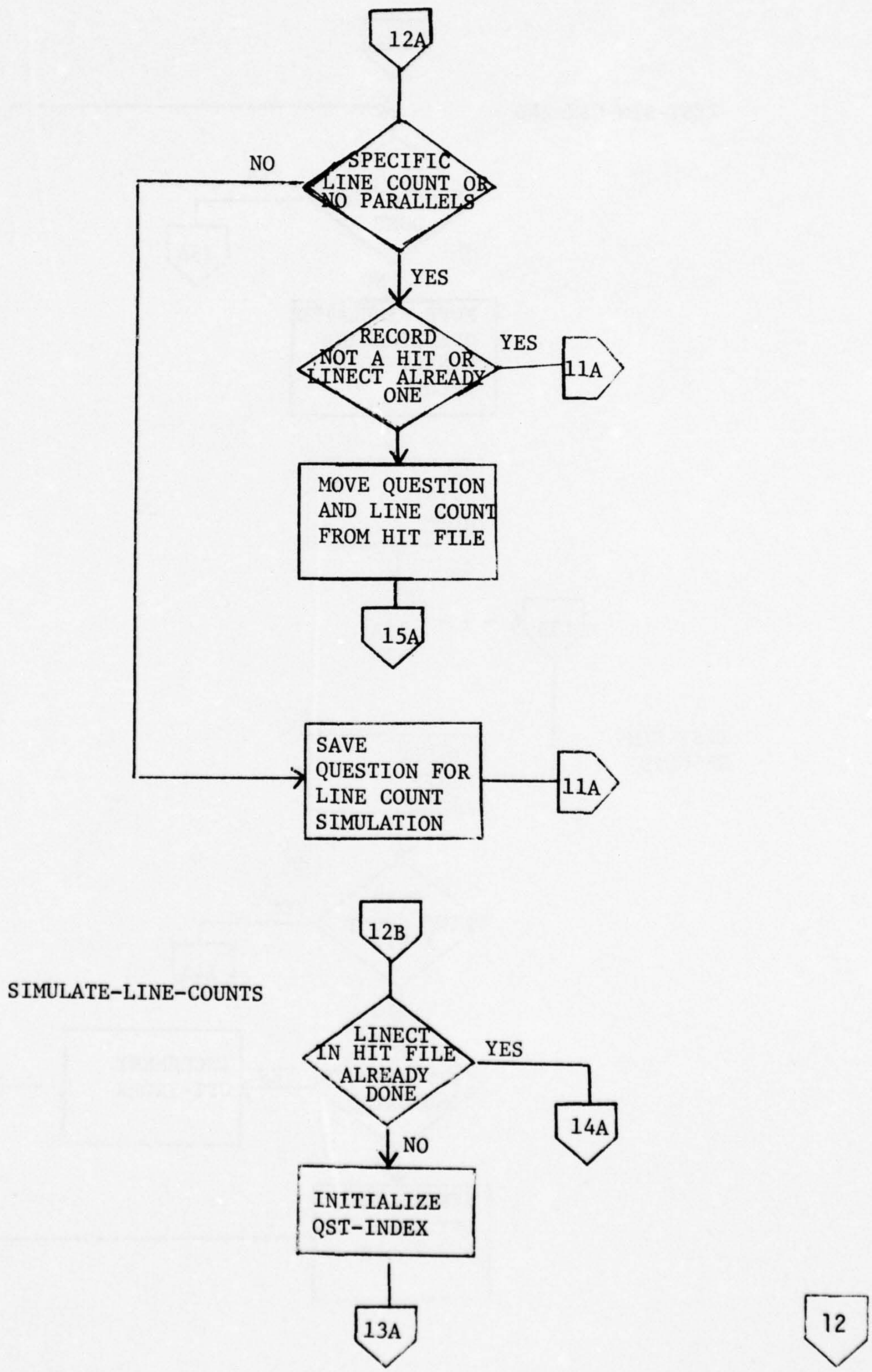

PERFORM-CONVERSION
8A
PERFORM CONVERSIONS THRU ITS EXIT
PERFORM BUILD - RESPONSE - RECORD
COMPLETE PACKET TO NTS
SET RETURN ADDRESS TO 4
THIS CAUSE CONTROL TO GO TO "TEST FOR REPEATERS" ON REENTRY OF ADS
RETURN TO ECM
8

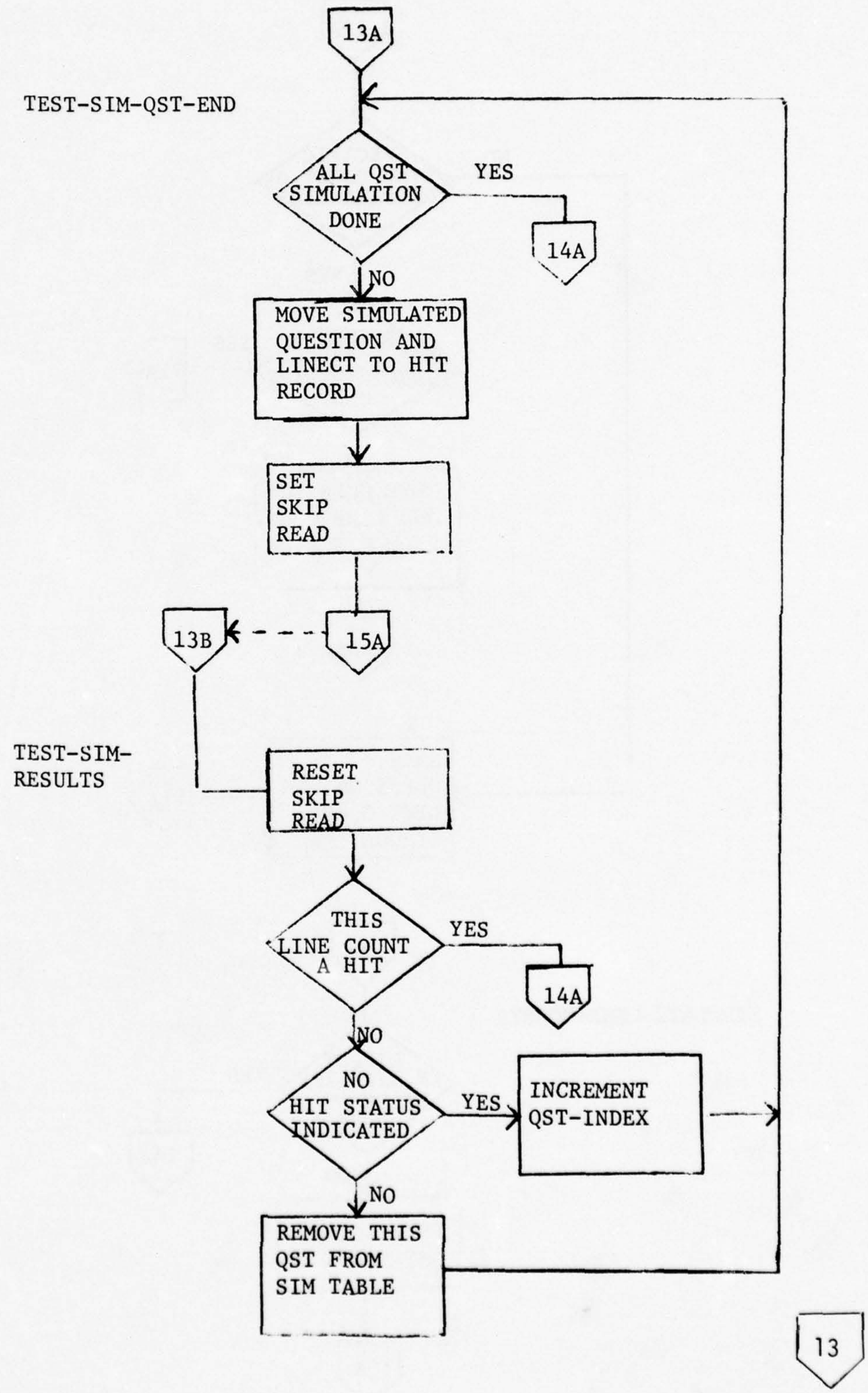

LIRC-DATA-RETRIEVAL
CLEAR LT-INDX AND TIMES-REPEATED
LT-INCREMENT
INCREMENT LT-INDX
LIRC TABLE FINISHED
YES
PUT OUT ERROR MESSAGE
RETURN WITH ERROR
NO
NOT PAR LIRC AND NOT FIRST TIME
YES
7B
NO
MATCH LIRCS WITH J TBL
NO
YES
SPECIFIC LINE COUNT NEEDED
NO
YES
COMPUTE ADJUSTED LT INDX
10A
9

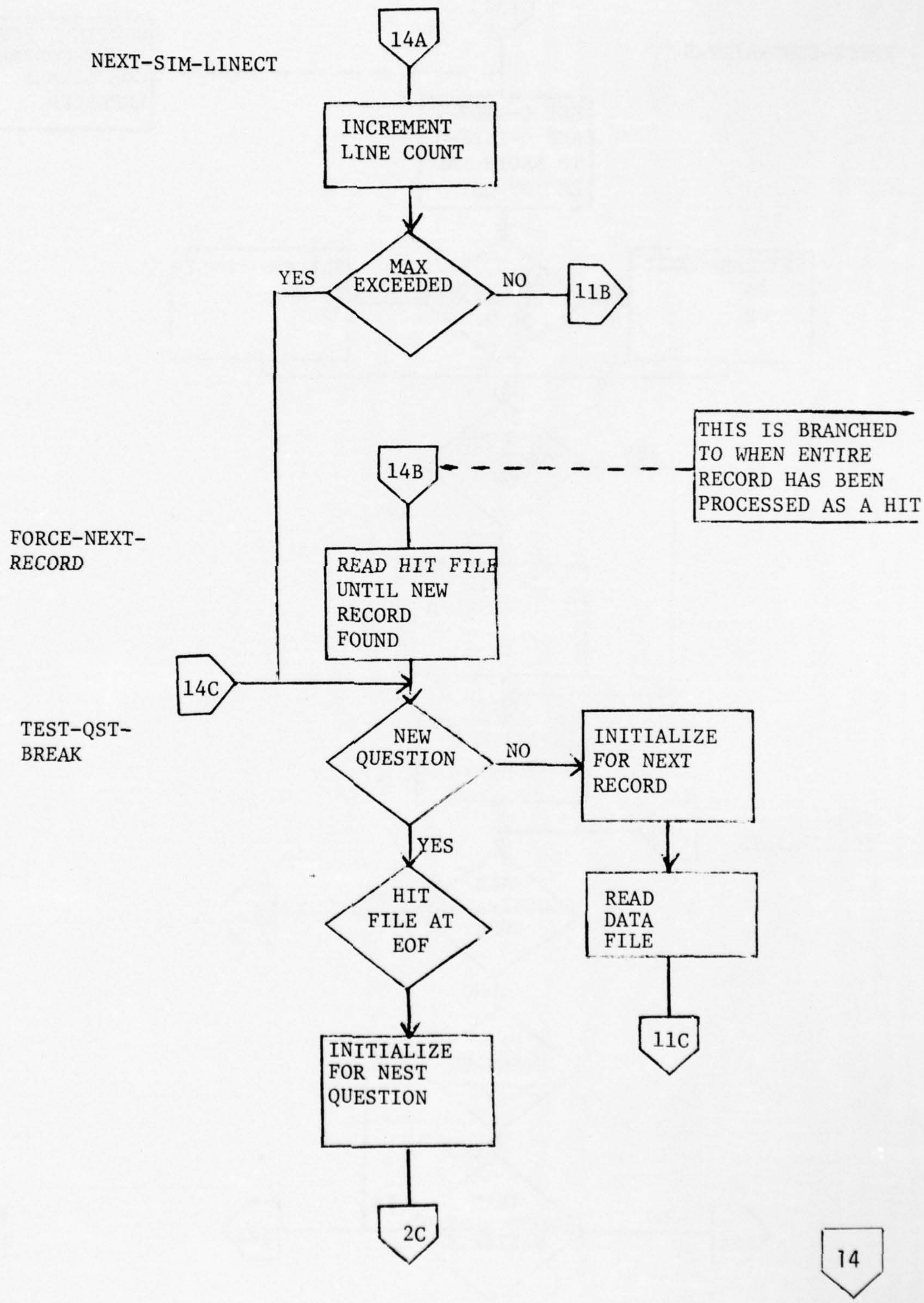

10A
BEGIN-RETRIEVAL
CLEAR OUTPUT PACKET
MOVE FIELD TO OUTPUT USING LIRC-TBL START AND LENGTH
RESULT SPACES AND NOT REPEATED
YES
7B
NO
LIRC-DATA-RETRIEVAL-EXIT
ROUTINE EXIT
10

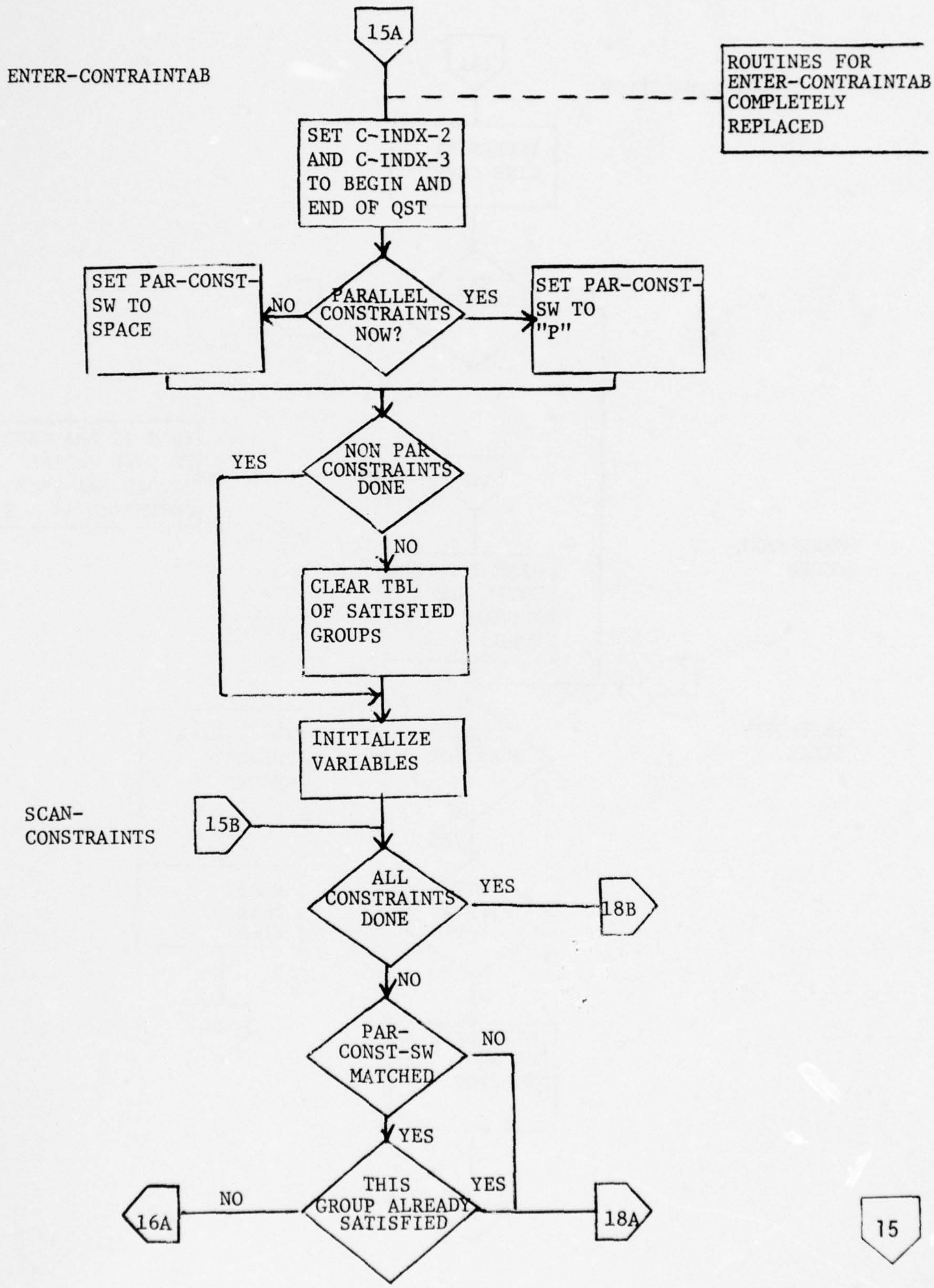
SHRM CHANGES TO ADS

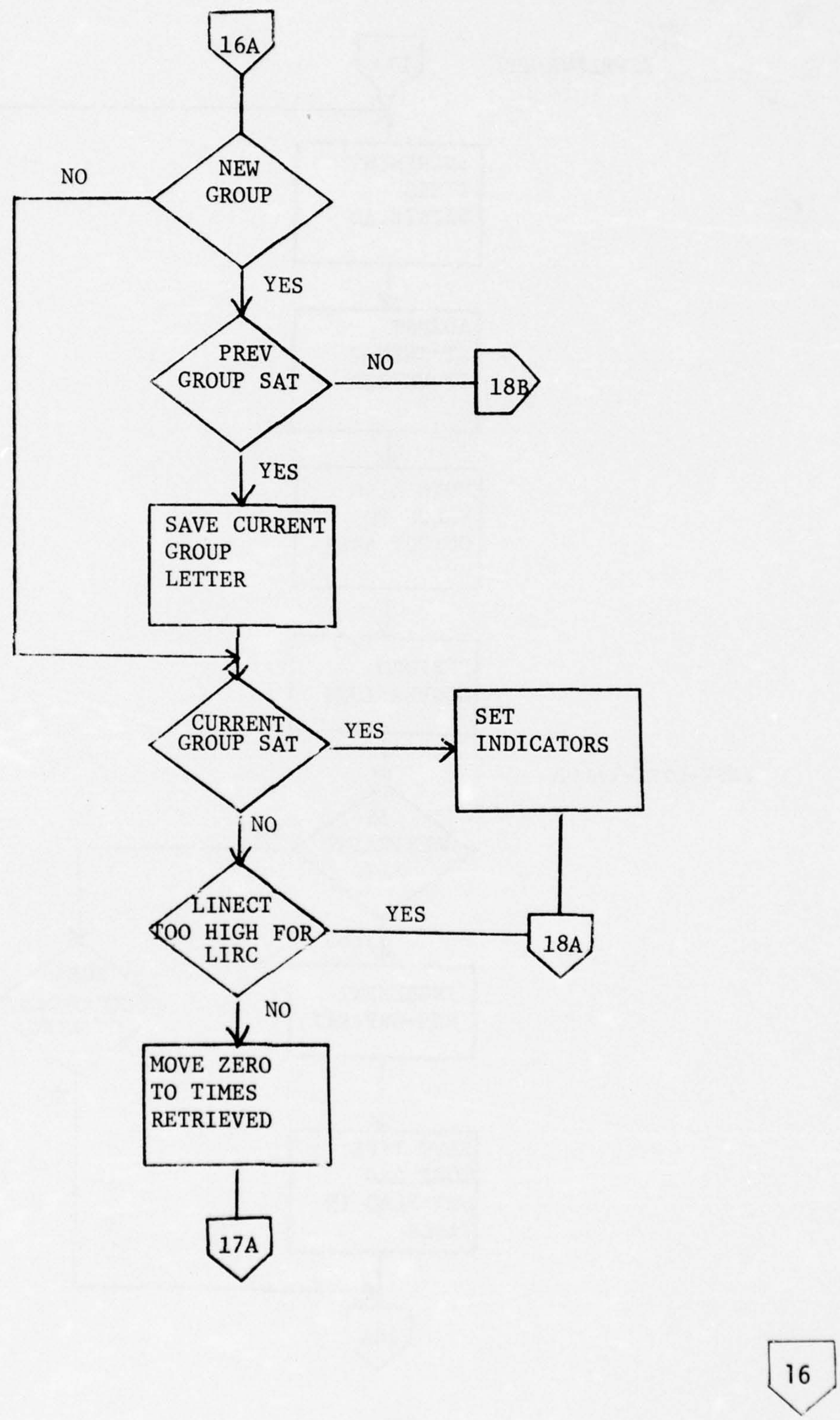


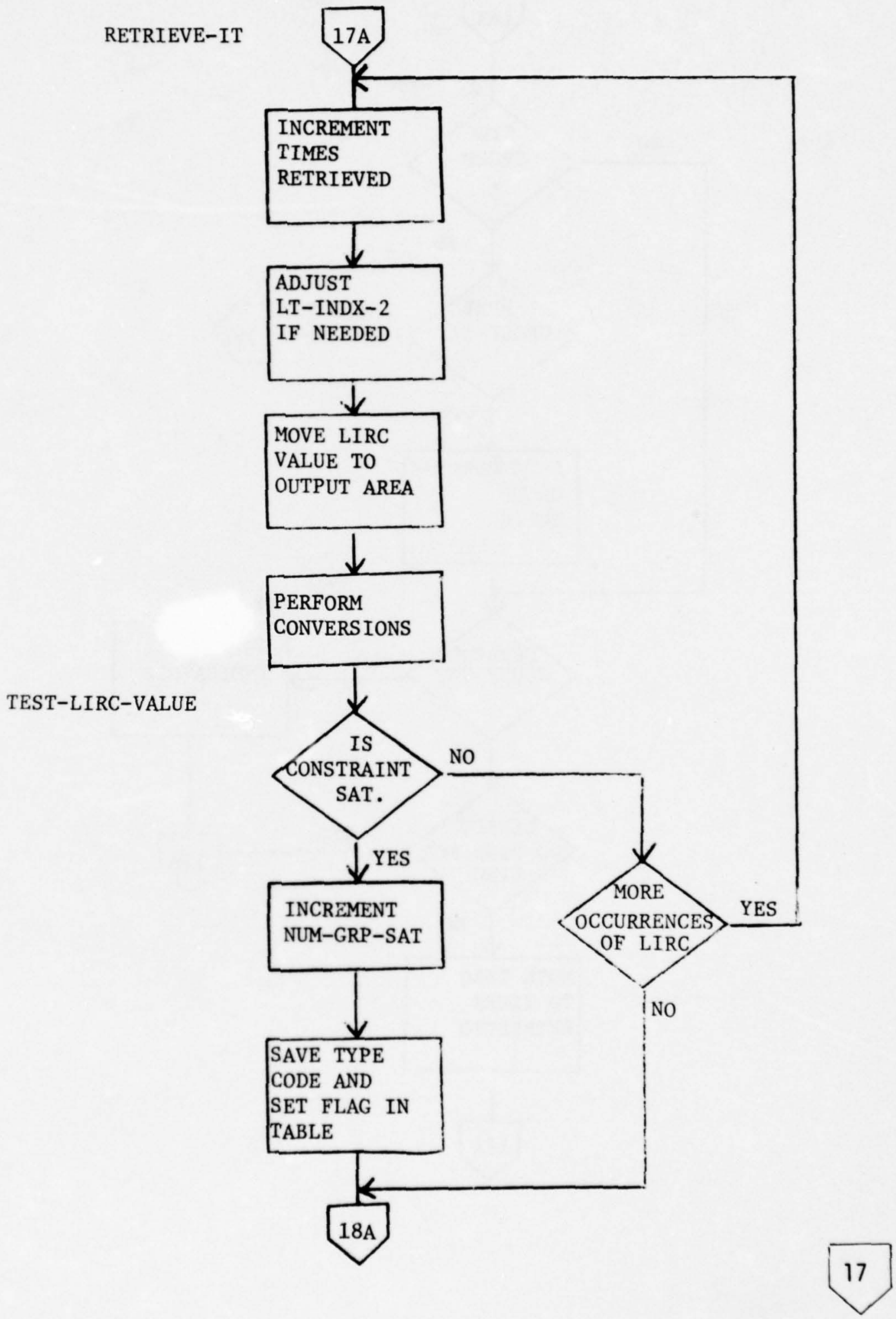

12A
SPECIFIC LINE COUNT OR NO PARALLELS
NO
YES
RECORD NOT A HIT OR LINECT ALREADY ONE
YES
11A
MOVE QUESTION AND LINE COUNT FROM HIT FILE
15A
SAVE QUESTION FOR LINE COUNT SIMULATION
11A
12B
SIMULATE-LINE-COUNTS
LINECT IN HIT FILE ALREADY DONE
YES
14A
NO
INITIALIZE QST-INDEX
13A
12

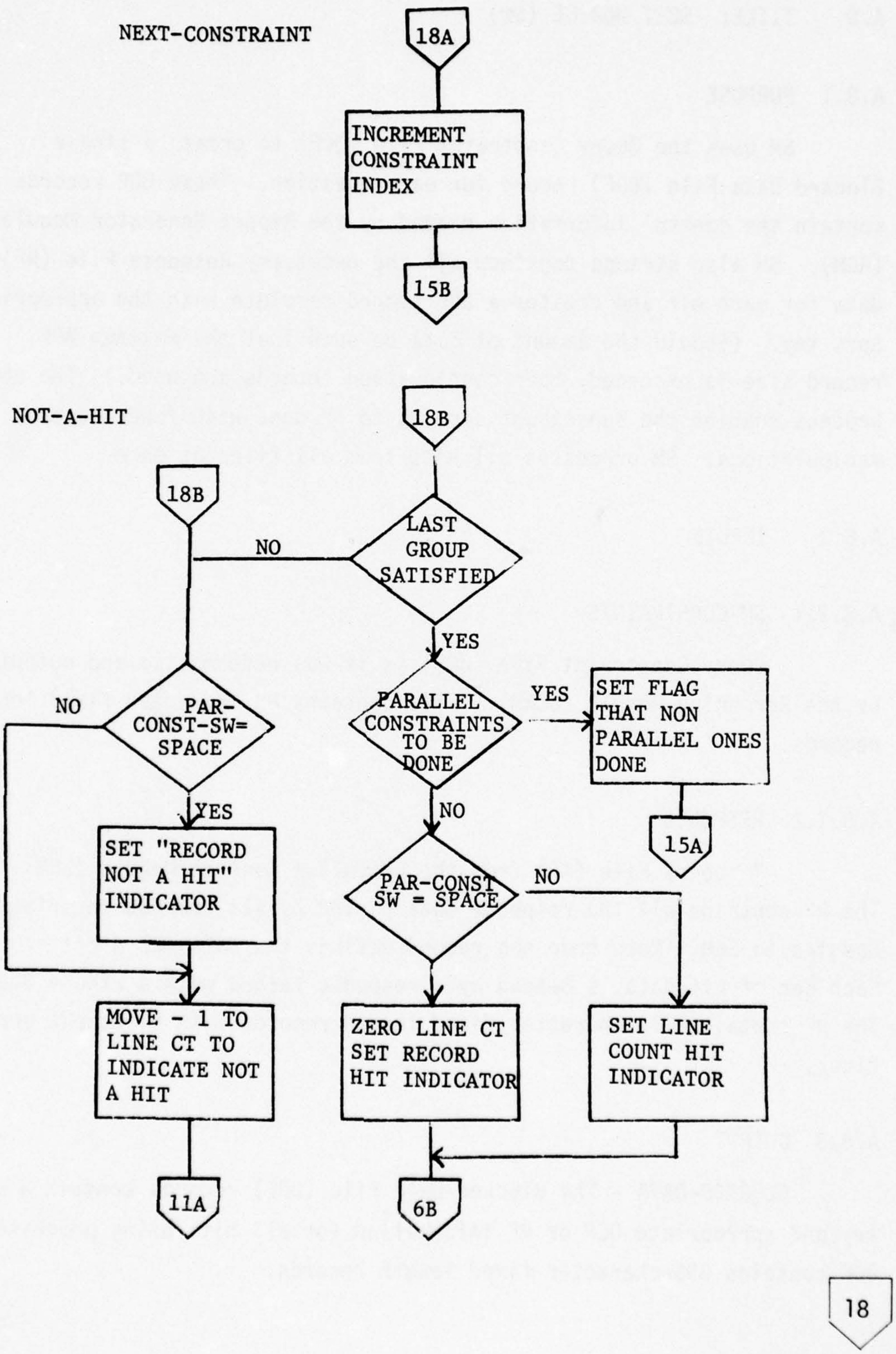

13A
TEST-SIM-QST-END
ALL QST SIMULATION DONE
YES
14A
NO
MOVE SIMULATED QUESTION AND LINECT TO HIT RECORD
SET SKIP READ
13B
15A
TEST-SIM-RESULTS
RESET SKIP READ
THIS LINE COUNT A HIT
YES
14A
NO
NO HIT STATUS INDICATED
YES
INCREMENT QST-INDEX
NO
REMOVE THIS QST FROM SIM TABLE
13


NEXT-SIM-LINECT
14A
INCREMENT LINE COUNT
MAX EXCEEDED
YES
NO
11B
THIS IS BRANCHED TO WHEN ENTIRE RECORD HAS BEEN PROCESSED AS A HIT
14B
FORCE-NEXT-RECORD
READ HIT FILE UNTIL NEW RECORD FOUND
14C
TEST-QST-BREAK
NEW QUESTION
NO
INITIALIZE FOR NEXT RECORD
YES
HIT FILE AT EOF
READ DATA FILE
11C
INITIALIZE FOR NEST QUESTION
2C
14


15A
ENTER-CONTRAINTAB
ROUTINES FOR ENTER-CONTRAINTAB COMPLETELY REPLACED
SET C-INDX-2 AND C-INDX-3 TO BEGIN AND END OF QST
SET PAR-CONST-SW TO SPACE
NO
PARALLEL CONSTRAINTS NOW?
YES
SET PAR-CONST-SW TO "P"
YES
NON PAR CONSTRAINTS DONE
NO
CLEAR TBL OF SATISFIED GROUPS
INITIALIZE VARIABLES
SCAN-CONSTRAINTS
15B
ALL CONSTRAINTS DONE
YES
18B
NO
PAR-CONST-SW MATCHED
NO
YES
16A
NO
THIS GROUP ALREADY SATISFIED
YES
18A
15


16A
NEW GROUP
NO
YES
PREV GROUP SAT
NO
18B
YES
SAVE CURRENT GROUP LETTER
CURRENT GROUP SAT
YES
SET INDICATORS
NO
LINECT TOO HIGH FOR LIRC
YES
18A
NO
MOVE ZERO TO TIMES RETRIEVED
17A
16


RETRIEVE-IT
17A
INCREMENT TIMES RETRIEVED
ADJUST LT-INDX-2 IF NEEDED
MOVE LIRC VALUE TO OUTPUT AREA
PERFORM CONVERSIONS
TEST-LIRC-VALUE
IS CONSTRAINT SAT.
NO
YES
INCREMENT NUM-GRP-SAT
MORE OCCURRENCES OF LIRC
YES
NO
SAVE TYPE CODE AND SET FLAG IN TABLE
18A
17



NEXT-CONSTRAINT
18A
INCREMENT CONSTRAINT INDEX
15B
NOT-A-HIT
18B
18B
NO
LAST GROUP SATISFIED
YES
NO
PAR-CONST-SW= SPACE
PARALLEL CONSTRAINTS TO BE DONE
YES
SET FLAG THAT NON PARALLEL ONES DONE
15A
YES
SET "RECORD NOT A HIT" INDICATOR
NO
PAR-CONST SW = SPACE
NO
MOVE - 1 TO LINE CT TO INDICATE NOT A HIT
ZERO LINE CT SET RECORD HIT INDICATOR
SET LINE COUNT HIT INDICATOR
11A
6B
18

A.8 TITLE: SORT MODULE (SM)

A.8.1 PURPOSE

SM uses the Query Constraint File (QCF) to create a single Blocked Data File (BDF) record for each question. These BDF records contain the control information needed by the Report Generator Module (RGM). SM also strings together all the necessary Response File (RF) data for each hit and creates a BDF record complete with the appropriate sort key. (Should the amount of data be such that the maximum BDF record size is exceeded, then continuation records are used.) The above process enables the subsequent sorting to be done with fewer record manipulations. SM processes all hits from all files at once.

A.8.2 INPUTS

A.8.2.1 SM-CONSTRAINTS

Query Constraint File (QCF) as it was reformatted and output by the Screening Module (SCRM). File contains 80-character fixed length records.

A.8.2.2 RESPONSE

Response File (RF) from the Executive Control Module (ECM). The RF contains all the response data passed by all the ADS Interface Modules to ECM. More than one record defines the data for a hit. Each set of hit data is headed by a response record with a LIRC = 0000. The RF contains 114-character fixed length records with 5 records per block.

A.8.3 OUTPUT

BLOCKED-DATA - The Blocked-Data File (BDF) records contain a sort key and appropriate QCF or RF information for all hits being processed. BDF contains 895-character fixed length records.

Section & Page 101.
Subsection
Date Documented
Change Notice #

Fold back at dotted line.

CARD OR RECORD LAYOUT - DOUBLE

SYSTEM ID: ☐ New ☐ Revision

RECORD TITLE:
FILE ID: BLOCKED DATA FILE
RECORD LENGTH: 23 MINIMUM ☑ FIXED ☐ 895 MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR 1 RECORD / BLOCK

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Positions 1–100:

1–2	3–4	5–6	7–8	9–100
TRAILERS (1)	LOC 1ST (2)	LOC LAST (3)	CONT CODE (4)	(5)

SORT KEY:
QUERY ID MAXSIZE = 3
QUESTION NO MAXSIZE = 3
DATA KEY MAXSIZE = 200
LIRE "0000" MAXSIZE = 15
TIE BREAKER MAXSIZE = 1
FILLER MAXSIZE = 7
DATA BLOCK (S): (DESCRIBED IN DETAIL BELOW)
NO. TRAILERS MAXSIZE = 2
DATA MAXSIZE = 102
FILLER MAXSIZE = 7

(1) "001" ... "110"
(2) NUMERIC
(3) NUMERIC
(4) 'X' OR SPACE
(5) ALPHANUMERIC

RECORD TITLE: DATA BLOCK B
FILE ID: BLOCKED DATA FILE
RECORD LENGTH: 3 MINIMUM ☑ FIXED ☐ 111 MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Positions 1–100:

1–2	3	4	5–6	7–8	9–14	15	16–18	19–24
TRAILERS (1)	FILLER	TYPE CODE (2)	SET NO (3)	FILLER	FACILITY ID (4)	FILLER	REPORT ID (5)	FILLER

(1) "03"
(2) "B"
(3) NUMERIC
(4) ALPHANUMERIC
(5) ALPHANUMERIC

Figure A-14

Section & Page 101.
Subsection
Date Documented
Change Notice #

Fold back at dotted line.

CARD OR RECORD LAYOUT - DOUBLE

SYSTEM ID: New ☐ Revision ☐

RECORD TITLE: DATA BLOCK C
FILE ID: BLOCKED DATA FILE
RECORD LENGTH: 3 MINIMUM ☑ FIXED ☐ 111 MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Positions 1–100:
TRAILERS (1) | FILLER (2) | TYPE CODE (3) | SET NO (4) | COMP CODE (5) | CONTROL IN (6) | LIRC 1 (7) | FORM 1 (8) | DECIMAL 1 (9) | ACRONYM 1 | LIRC 2 | FORM 2 | DECIMAL 2 | ACRONYM 2

(1) '03' OR '05'
(2) 'C'
(3) NUMERIC
(4) 'A' --- 'H'
(5) 'K', 'N' OR 'P'
(6) NUMERIC
(7) 'I', 'R', 'N', 'A', 'L' OR SPACE
(8) NUMERIC OR SPACE
(9) ALPHANUMERIC

RECORD TITLE: DATA BLOCK H
FILE ID: BLOCKED DATA FILE
RECORD LENGTH: 3 MINIMUM ☑ FIXED ☐ 111 MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Positions 1–100:
TRAILERS (1) | FILLER (2) | TYPE CODE (3) | SET NO | FILLER | HEADING DATA (4) | FILLER

(1) '09'
(2) 'H'
(3) NUMERIC
(4) ALPHANUMERIC

Figure A-15

Section & Page 101.
Subsection
Date Documented
Change Notice #

Fold back at dotted line.

SYSTEM ID:

New ☐ Revision ☐

CARD OR RECORD LAYOUT - DOUBLE

RECORD TITLE: DATA BLOCK J
FILE ID: BLOCKED DATA FILE

RECORD LENGTH: 3 MINIMUM ☑ FIXED ☐ 111 MAX.

WORDS ☐ BYTES ☐ CHARACTERS ☑

BLOCKING FACTOR

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

1–2	3	4	5–6	7–8	9–12	13–14	15–16	17–100
TRAILERS	NO PRINT	TYPE CODE	SET NO	FILLER	LIRC	FILLER	CHAR CNT	DATA VALUE OCCURS 1 TO 96 TIMES DEPENDING ON CHAR CNT

RECORD TITLE: DATA BLOCK J (CONTINUED)
FILE ID:

RECORD LENGTH: MINIMUM ☐ FIXED ☐ MAX.

WORDS ☐ BYTES ☐ CHARACTERS ☐

BLOCKING FACTOR

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

1–12	13–19	20–100
	FILLER	

Figure A-16

A.8.4 INTERMEDIATE

SORT-FILE - The Sort File is used to sort the BDF before it is used by the Report Generator Module (RGM). Sort File contains 895-character fixed length records.

A.8.5 RECORD FORMATS - See Figures A1-A5, A7, A14-A16

A.8.6 STORAGE REQUIREMENTS

Program (less library modules) occupies 22,000 bytes on UNIVAC Series 70/45.

A.8.7 PROCESSING DESCRIPTION (Refer to SM Flowchart, page 175)

A.8.7.1 GENERAL DESCRIPTION

The Sort Module (SM) begins processing by reading the Query Constraint File (QCF) that was output by the Screening Module (SCRM). Whenever a new question* is encountered, SM checks to see if the user has requested the question be deleted or the question had no hits. In either case, SM reads the QCF until it encounters the next question. Then all the necessary indicators for the new question are reset.

Next, each QCF record of the question is processed according to its type code. Type A records will be used to set the merge indicator. Type B, C, and H records will be used to create a control record on the Blocked Data File (BDF) for the question. Type D records will be used to create sort LIRC data. After all QCF records for the question have been processed, SM checks the last D LIRC to see if it was 0000. If not, an additional sort LIRC is created for the question. Then the BDF control record is written. The remaining questions are processed in this same manner.

---

* For the purposes of this writing, "question" will refer to either a single, non-merged question, or a set of merged questions.

When all questions have been processed, SM creates BDF data records for all the hit data on the Response File (RF). Then all the BDF records are sorted and control is returned to the Executive Control Module (ECM).

A.8.7.2 QCF TYPE CODE PROCESSING

(a) A Records - If type code for a QCF record is A, the merge question indicator is set to 1 (on) or 0 (off), whichever is appropriate.

(b) B, C, and H records - If type code for a QCF record is B, C, or H, data is moved from the QCF record to the BDF control record being created for the question. Should there be so many B, C, and H records for a question that the maximum size of the BDF record would be exceeded, SM will ignore the extra records. Only one BDF control record is created for each question. The sort key of the control record will be the question number and one trailer (8 characters) of low-values.

Also, the C LIRCs will be stored in a table for later use. The C LIRC table will handle a maximum of 20 entries. If a C record was not used in the creation of the BDF control record, it will not be put in the C LIRC table. Should there be more than 20 eligible C LIRCs, the excess will be ignored.

(c) D Records - If type code for a QCF record is D (indicating a sort LIRC), then the D record information is moved to an entry in the D data table in Working-Storage. This table will hold 21 entries, however, only 20 are reserved for D record data. The additional entry is for LIRC 0000 if it was not on the last D record processed for the question. All questions will sort on LIRC 0000 as the final sort LIRC. Should more than 20 D records occur for one question, the excess will be ignored.

Before adding an entry to the D data table, SM will check that the last entry did not cause the maximum possible sort key size for the question to equal or exceed 200. The maximum possible sort key size is computed by keeping a running total of all the character count fields of the D LIRCs for the question as they are being processed. The actual maximum sort key size on the UNIVAC is 255, but SM will allow no more D LIRCs (except LIRC 0000) after a LIRC has caused the maximum possible sort key size to be 200 or greater. Since a COBOL Sort is being used, if the actual sort key size for any BDF record should exceed 255, the extra sort data will be ignored in the sort.

A.8.7.3 BLOCK RESPONSE DATA

(a) General Description - To create the BDF data records, SM reads the first RF record of a hit. (Note: Each set of hit records begins with a LIRC 0000 record.) If the hit being processed belongs to a question which the user wants deleted, then the RF is read until the first record of a hit for a question which is not being deleted is encountered. Then SM processes all the sort RF records for the hit. These records are used to create the sort key for the hit. Next, the print RF records on the hit are used to create the data trailers for the BDF record. After all the data for a hit have been processed, the BDF record is written and the next set of hit data is processed. This continues until an end-of-file is reached on the RF.

(b) Process Sort Response Records - The sort RF records are those records for which the sort indicator field is equal to space. To create the BDF record sort key for a hit, all the sort RF records for the hit are retrieved and

stored in a sort response table. This table will hold a maximum of 21 entries. Should more than 21 entries occur, the excess will be ignored. As each record is stored, it is checked to see if it will also be used as print RF data. To determine this, SM checks the no-print field. If it is a space, then the sort LIRC is to be printed. If not, then SM determines whether the sort LIRC is also a compute LIRC by searching the C LIRC table. If so, then this sort LIRC will also be treated as a print LIRC. To keep track of the sort LIRCs which are also print LIRCs, SM stores both the index value from the sort response table of the sort LIRC and the LIRC itself in a hold data table.

After all the SF records for a hit have been stored, the sort key is created. This is done in a Working-Storage area because the sort key will be needed again if there are continuation BDF records. Also, its size (less the Question Number field) will be a multiple of eight since the trailer fields of the BDF are eight characters long and sort key data and print data cannot occur in the same trailer. The sort key is created by retrieving, in order, each D LIRC from the D data table and matching it with a LIRC from the sort response table. When a match is found, the data value from the sort response entry is moved to the next characters of the sort key area. Should a match not be found, then two characters are moved to the sort key. These characters are space and low-value for a low sort, and two high-values for a high sort. (Note: If the maximum data size for the LIRC is one, then only the first character is moved to the key.) After the last D LIRC has been matched, SM checks to see if the actual size of the sort key for

the hit is greater than the previous largest key for any other hit (regardless of the question). If so, the new key size becomes the largest key. Then the remaining characters (if any) of the last sort trailer are low-value filled and the sort key in Working-Storage is moved by trailers to the BDF record.

(c) Move Sort Data - When a match has been found in the sort response table for the D LIRC being processed, the data value from the sort response entry will be moved according to the following procedures. First, the D data entry is checked to see if the sort is prefix or partial. If it is a prefix sort, the start indicator is set to zero and the stop indicator is set to the number of prefix characters desired. The start indicator is actually the initial setting of the character index (CHAR) used to move the sort response data value. It indicates how many characters are to be skipped before characters are retrieved. The stop indicator (MOVE-STOP) indicates the last possible position of the last character to be retrieved. For a partial sort, the start indicator is set to the number of characters skipped before the partial sort. The stop indicator equals the sum of the start indicator and the number of characters in the partial sort. If the sort is neither prefix nor partial, then the start indicator is set to zero and the stop indicator is set to the maximum data size.

Next, the D data entry is checked for the type of sort. If it is a low sort, then the characters as they occur in the data value are moved to the sort key area. If, however, the user requests a high sort for the LIRC, each character is moved to the last position of a four-character alphanumeric data item that is redefined as a computational (COMP) item. Then the COMP item is subtracted from 250. The last alphanumeric character of the result of the computation is the "complement" character of the original character. This

complement character is moved to the sort key. (Note: Since some LIRCs may be sorted low and others high in the same key, this complement procedure is needed for the high sorts.)

The sort data characters or complement characters are moved by incrementing the character index until either the stop indicator or the sort response data size is reached. When one of these occurs, if the character index is less than 90 and the sort response data size is less than the stop indicator, then a space and a low-value for a low sort, or two high-values for a high-sort, are moved to the next sort key characters. The moving for that LIRC is complete. If the stop indicator equals the sort response data size, the move is complete. If, however, the character index is greater than or equal to 90, but not equal to the stop indicator, then the next sort response table entry is retrieved. If its LIRC equals the one being processed, then the moving process is continued until either the stop indicator or the end of the sort response value is reached. Before continuing the moving, the start indicator is set to zero and the stop indicator equals the value of the previous one less the character index (before it is reset). If, however, the LIRCs were not equal, then a space and a low-value for low sort or two high-values for a high sort are moved to the next sort key characters to complete that portion of the key.

(d) Process Print Responses - After the sort key has been created for a hit, the data to be printed, or used in computations, is processed. The sort RF records already processed for the current hit were in order sort by LIRC. Therefore, the hold data table, which is a subset of the sort RF records, is also in LIRC order. The print RF records (those for which the sort indicator equals 9) are also in LIRC order and occur immediately after the sort RF records for the hit. The BDF trailers of data information are created in LIRC order by merging the LIRC

data from the hold table and the print RF records. If SM determines that the next entry from the hold data table precedes the next RF record, then the index in the hold data entry is used to retrieve the appropriate data from the sort response table. This data is then used to create the next data trailers. If, however, the next RF record comes first, then the next data trailers are created with the information from it.

Before the data trailers are added to the BDF record, SM computes how many trailers will be needed for the LIRC being processed. If all the trailers for a LIRC will not fit on the current BDF record, then none are put on it. The continuation field of the BDF record is set, the remaining necessary fields are filled in, and the record is written. Then a continuation record is set up using the same sort key with the next highest tie breaker character. The data trailers for the current LIRC are then added to the new record.

After all hold data and print RF records for a hit have been processed, SM determines whether there are any trailers on the current BDF record. If not, nothing is written and the processing continues with the creation of the sort key for the next hit. If there are trailers on the current BDF record, then the number of trailers is compared to the maximum number of sort trailers for the question. If the maximum number of sort trailers is greater, then enough space-filled trailers are added to the current BDF record to give it as many trailers as the maximum number of sort trailers for the question. The remaining necessary fields of the record are filled in and the completed BDF record is written. Print response processing for the hit is complete. SM is now ready to process the next hit.

A.8.8 COMMENTS ON SM

The maximum sort key size on the UNIVAC is 255 characters. The sort key on the BDF record will be composed of the last six characters of the fixed portion of the record (question number) and a variable number of eight-character trailers. Some of the pending Program Change Proposals use 254 as the maximum key size. This is because 255 would create the possibility of a sort trailer which could never have more than one character of data.

Whenever a sort value for a LIRC is shorter than the maximum size (or not present at all), SM adds space, low value for low sorts or two high-values for high sorts to the end of the sort value. The next sort value is then placed immediately after the low-value or the high-values. This process enables the sort keys to be made compact. Space, low-value was chosen for the low sort instead of two low-values so there will be no possibility of any BDF data records sorting before the BDF control record for the same question.

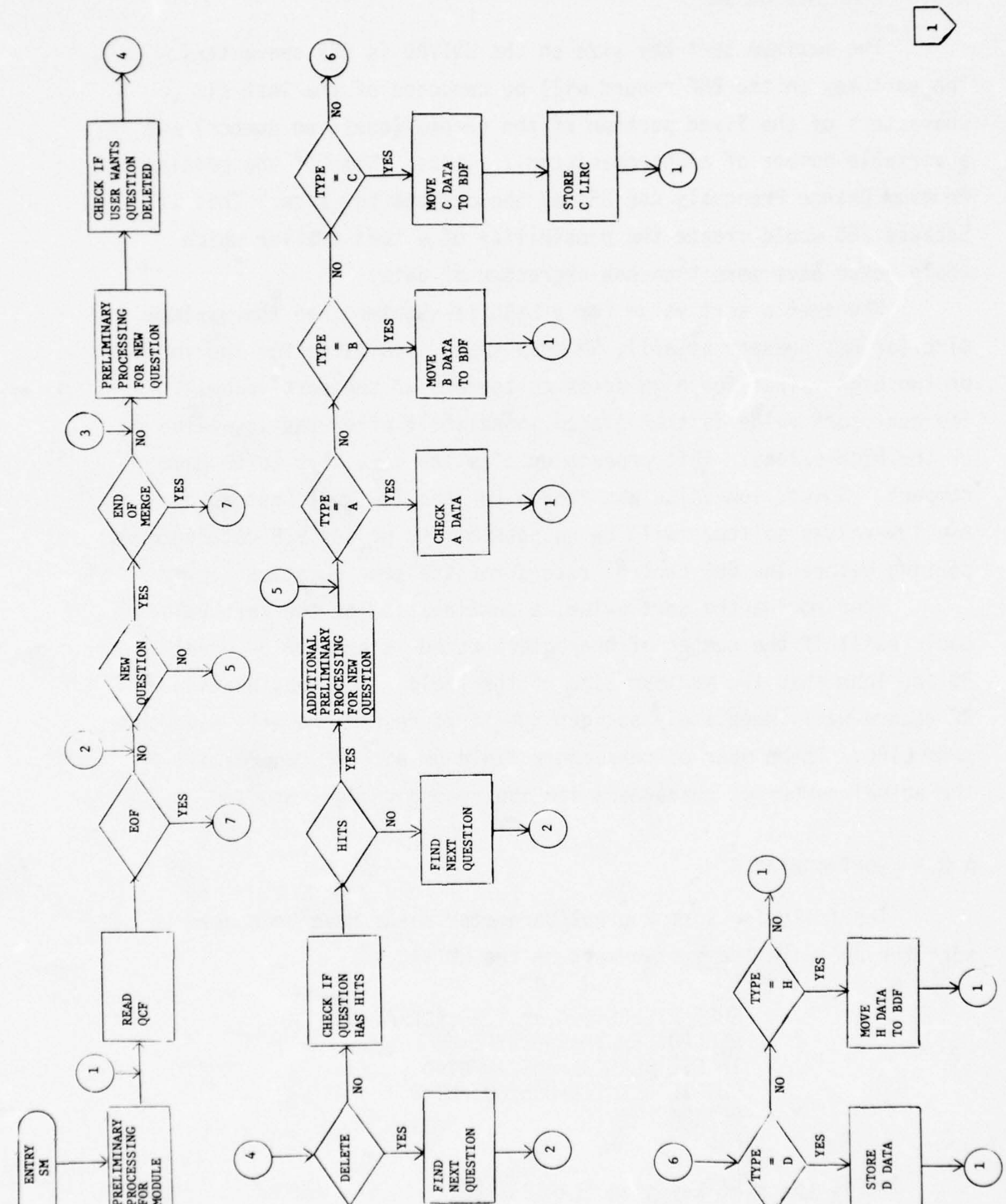
When moving the sort value, a continuation of the sort value could exist if the number of characters moved is greater or equal to 90 and less than the maximum size of the field. The continuation RF record will immediately succeed the first record and will have the same LIRC. The number-of-characters field on each RF record will be the actual number of characters for the record being processed.

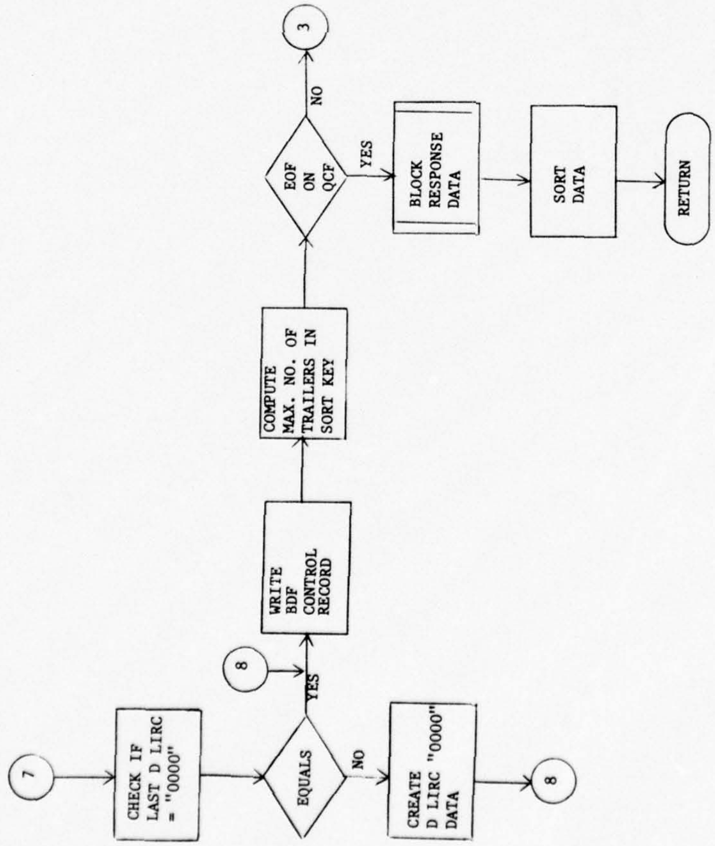
A.8.9 SORT CONTROLS

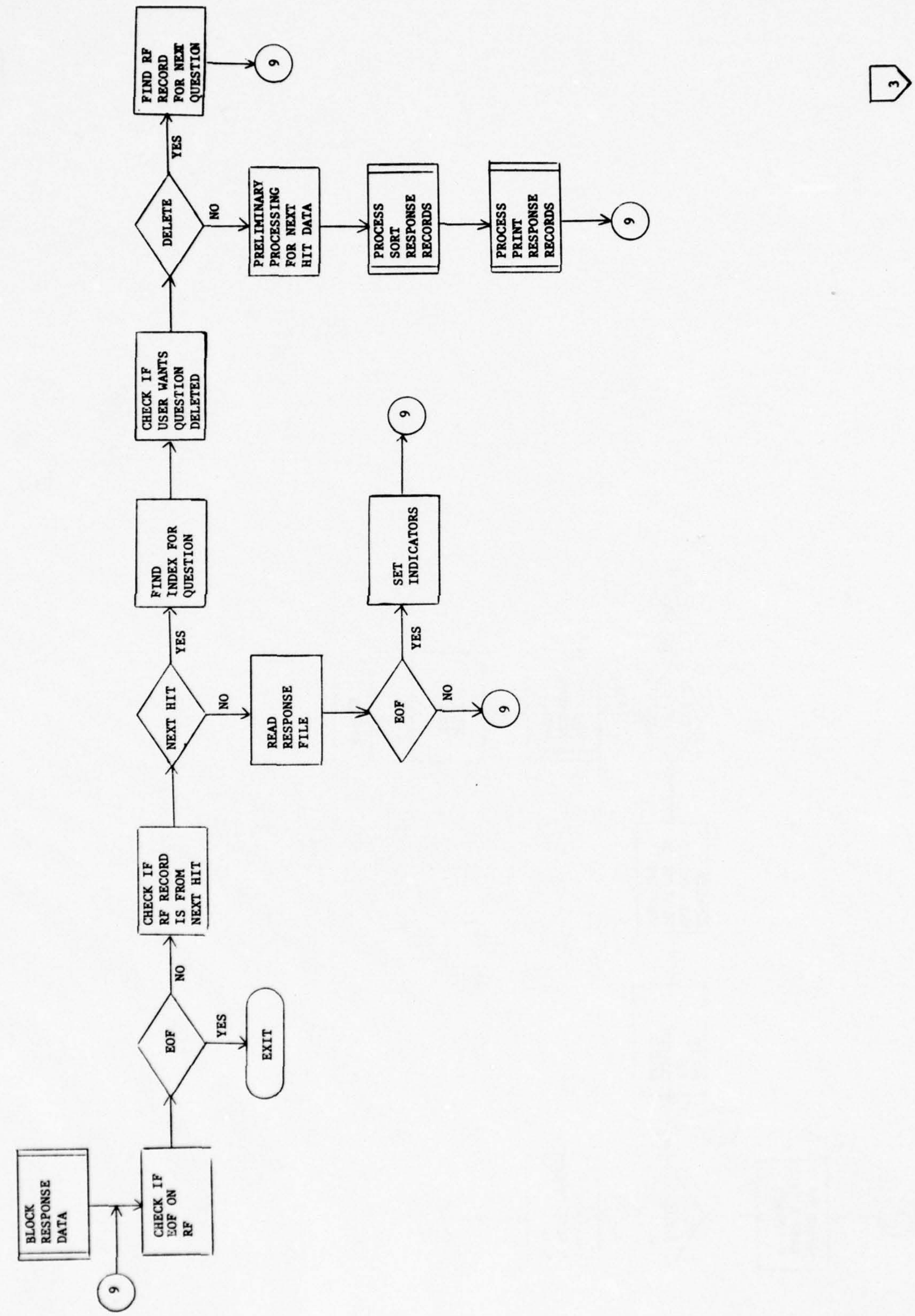
The following sort control parameter cards have been used to sort the BDF with the system sort on the UNIVAC.

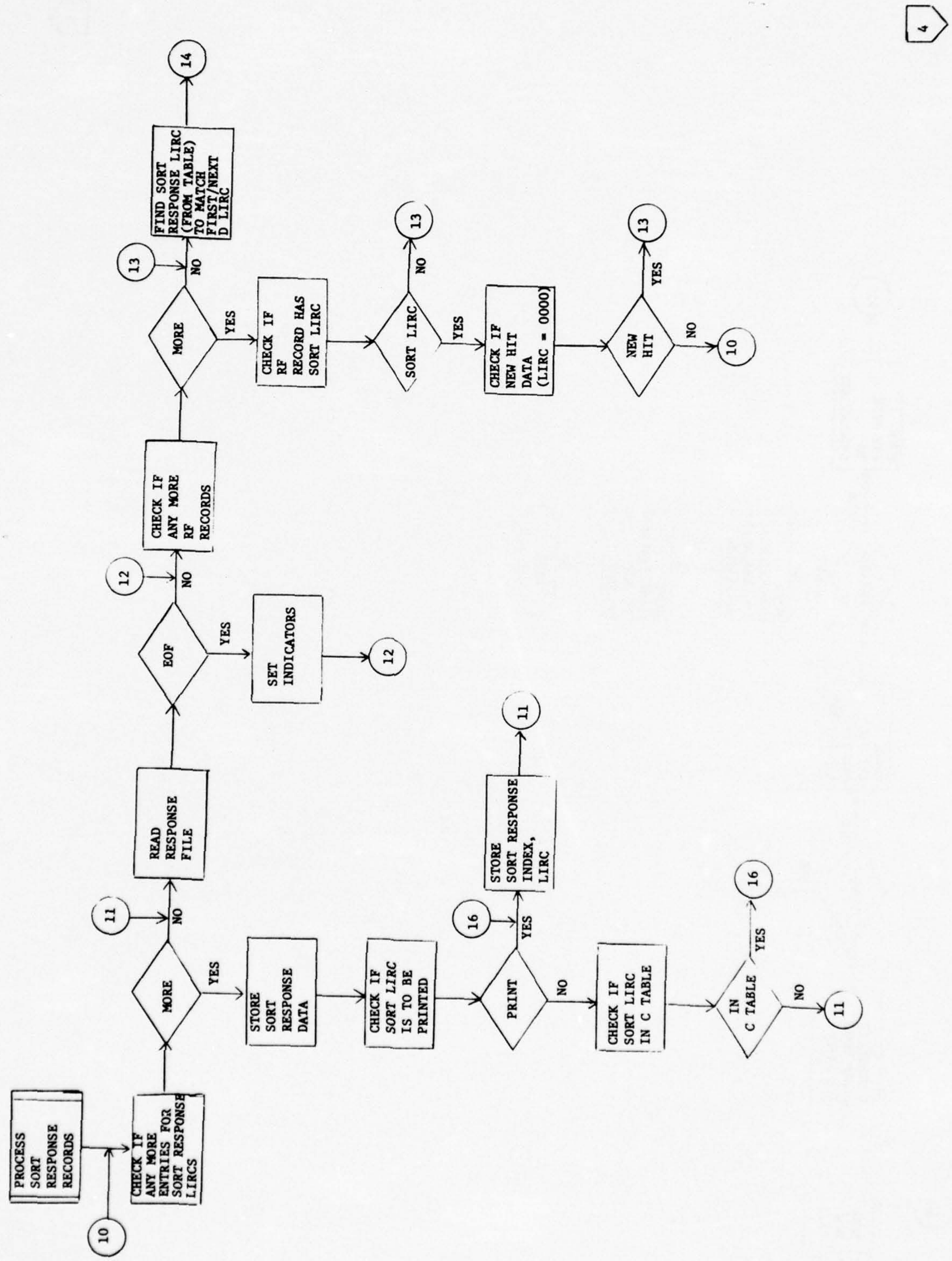
```
SORT FIELDS=(10,n*,1,A),FORMAT=BI
RECORD LENGTH=895,TYPE=F
INPFIL BLKSIZE=895,INPUT=D
OUTFIL BLKSIZE=895,OUTPUT=D
OPTION PRINT
END
```

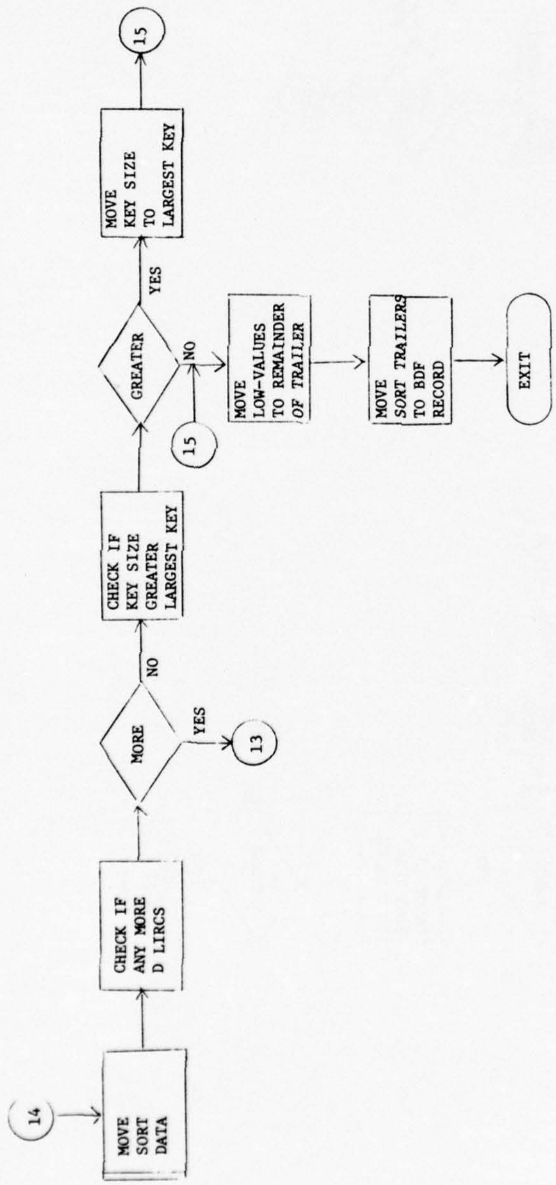
*n is the sort key size (LARGEST-KEY)

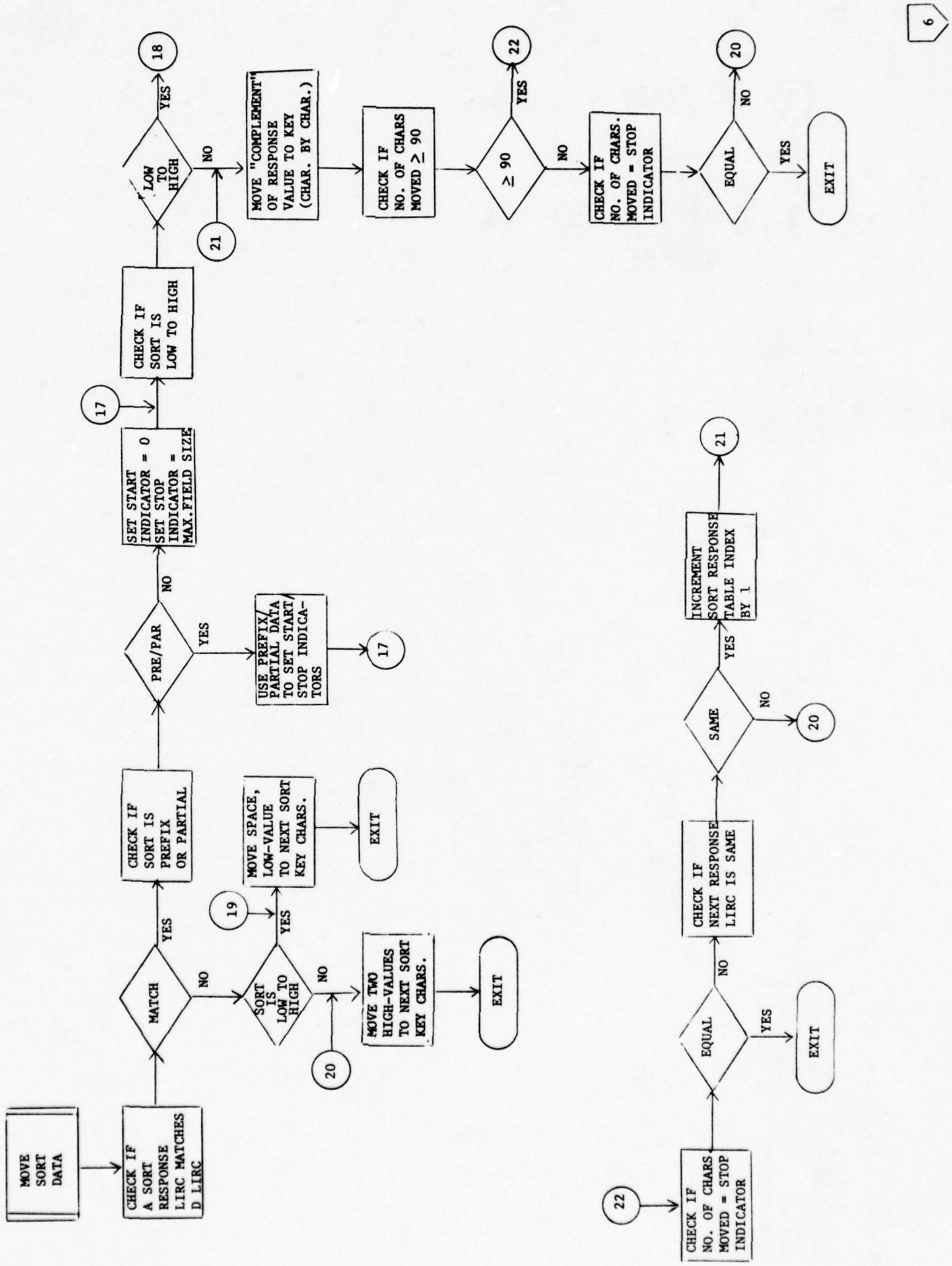

ENTRY SM
PRELIMINARY PROCESSING FOR MODULE
READ QCF
EOF
NEW QUESTION
END OF MERGE
PRELIMINARY PROCESSING FOR NEW QUESTION
CHECK IF USER WANTS QUESTION DELETED
DELETE
FIND NEXT QUESTION
CHECK IF QUESTION HAS HITS
HITS
FIND NEXT QUESTION
ADDITIONAL PRELIMINARY PROCESSING FOR NEW QUESTION
TYPE = A
CHECK A DATA
TYPE = B
MOVE B DATA TO BDF
TYPE = C
MOVE C DATA TO BDF
STORE C LIRC
TYPE = D
STORE D DATA
TYPE = H
MOVE H DATA TO BDF
YES
NO
1
2
3
4
5
6
7

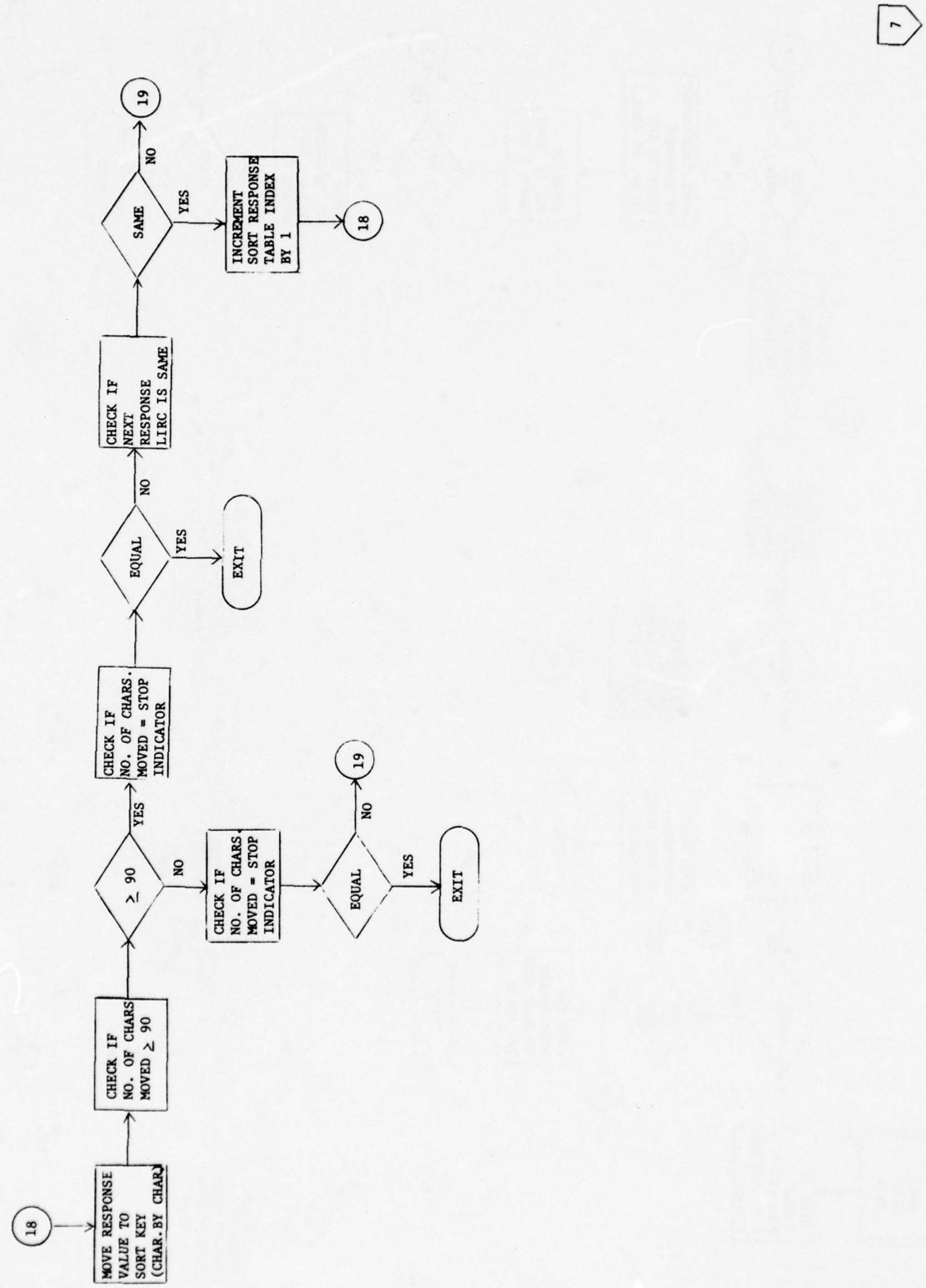

7
CHECK IF LAST D LIRC = "0000"
EQUALS
YES
NO
8
CREATE D LIRC "0000" DATA
8
WRITE BDF CONTROL RECORD
COMPUTE MAX. NO. OF TRAILERS IN SORT KEY
EOF ON QCF
NO
3
YES
BLOCK RESPONSE DATA
SORT DATA
RETURN


BLOCK RESPONSE DATA
9
CHECK IF EOF ON RF
EOF
YES
EXIT
NO
CHECK IF RF RECORD IS FROM NEXT HIT
NEXT HIT
YES
FIND INDEX FOR QUESTION
CHECK IF USER WANTS QUESTION DELETED
DELETE
YES
FIND RF RECORD FOR NEXT QUESTION
9
NO
PRELIMINARY PROCESSING FOR NEXT HIT DATA
PROCESS SORT RESPONSE RECORDS
PROCESS PRINT RESPONSE RECORDS
9
NO
READ RESPONSE FILE
EOF
NO
9
YES
SET INDICATORS
9
3

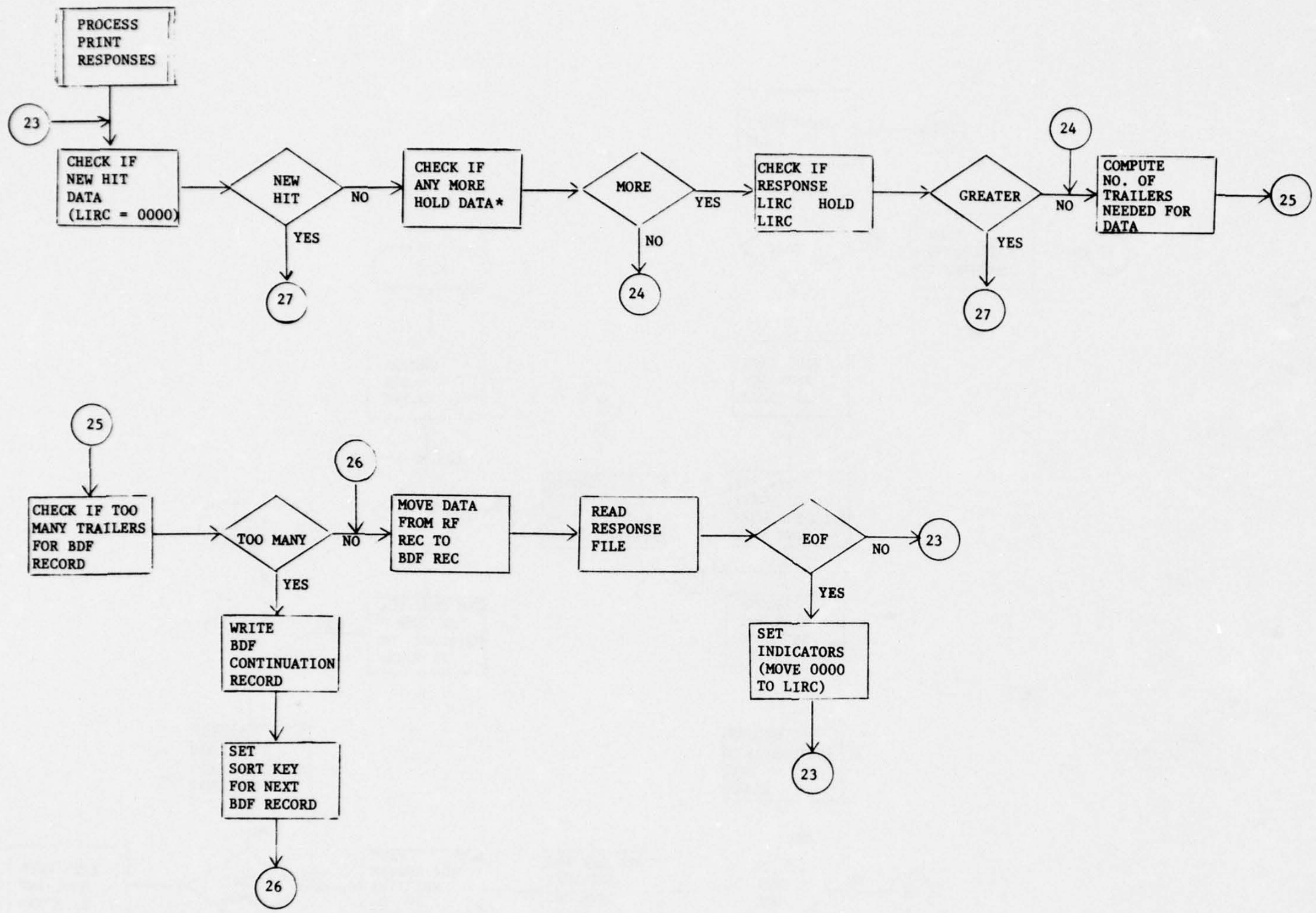

PROCESS SORT RESPONSE RECORDS
10
CHECK IF ANY MORE ENTRIES FOR SORT RESPONSE LIRCS
MORE
YES
NO
11
STORE SORT RESPONSE DATA
CHECK IF SORT LIRC IS TO BE PRINTED
PRINT
16
YES
STORE SORT RESPONSE INDEX, LIRC
11
NO
CHECK IF SORT LIRC IN C TABLE
IN C TABLE
YES
16
NO
11
READ RESPONSE FILE
EOF
YES
SET INDICATORS
12
NO
12
CHECK IF ANY MORE RF RECORDS
MORE
YES
CHECK IF RF RECORD HAS SORT LIRC
SORT LIRC
NO
13
YES
CHECK IF NEW HIT DATA (LIRC = 0000)
NEW HIT
YES
13
NO
10
NO
13
FIND SORT RESPONSE LIRC (FROM TABLE) TO MATCH FIRST/NEXT D LIRC
14
4


14
MOVE SORT DATA
CHECK IF ANY MORE D LIRCS
MORE
YES
13
NO
CHECK IF KEY SIZE GREATER LARGEST KEY
GREATER
YES
MOVE KEY SIZE TO LARGEST KEY
15
NO
15
MOVE LOW-VALUES TO REMAINDER OF TRAILER
MOVE SORT TRAILERS TO BDF RECORD
EXIT


MOVE SORT DATA
CHECK IF A SORT RESPONSE LIRC MATCHES D LIRC
MATCH
YES
NO
CHECK IF SORT IS PREFIX OR PARTIAL
PRE/PAR
YES
NO
USE PREFIX/ PARTIAL DATA TO SET START/ STOP INDICA-TORS
17
SET START INDICATOR = 0 SET STOP INDICATOR = MAX.FIELD SIZE
17
CHECK IF SORT IS LOW TO HIGH
LOW TO HIGH
YES
18
NO
21
MOVE "COMPLEMENT" OF RESPONSE VALUE TO KEY (CHAR. BY CHAR.)
CHECK IF NO. OF CHARS MOVED ≥ 90
≥ 90
YES
22
NO
CHECK IF NO. OF CHARS. MOVED = STOP INDICATOR
EQUAL
NO
20
YES
EXIT
SORT IS LOW TO HIGH
19
YES
MOVE SPACE, LOW-VALUE TO NEXT SORT KEY CHARS.
EXIT
NO
20
MOVE TWO HIGH-VALUES TO NEXT SORT KEY CHARS.
EXIT
22
CHECK IF NO. OF CHARS MOVED = STOP INDICATOR
EQUAL
YES
EXIT
NO
CHECK IF NEXT RESPONSE LIRC IS SAME
SAME
NO
20
YES
INCREMENT SORT RESPONSE TABLE INDEX BY 1
21
6

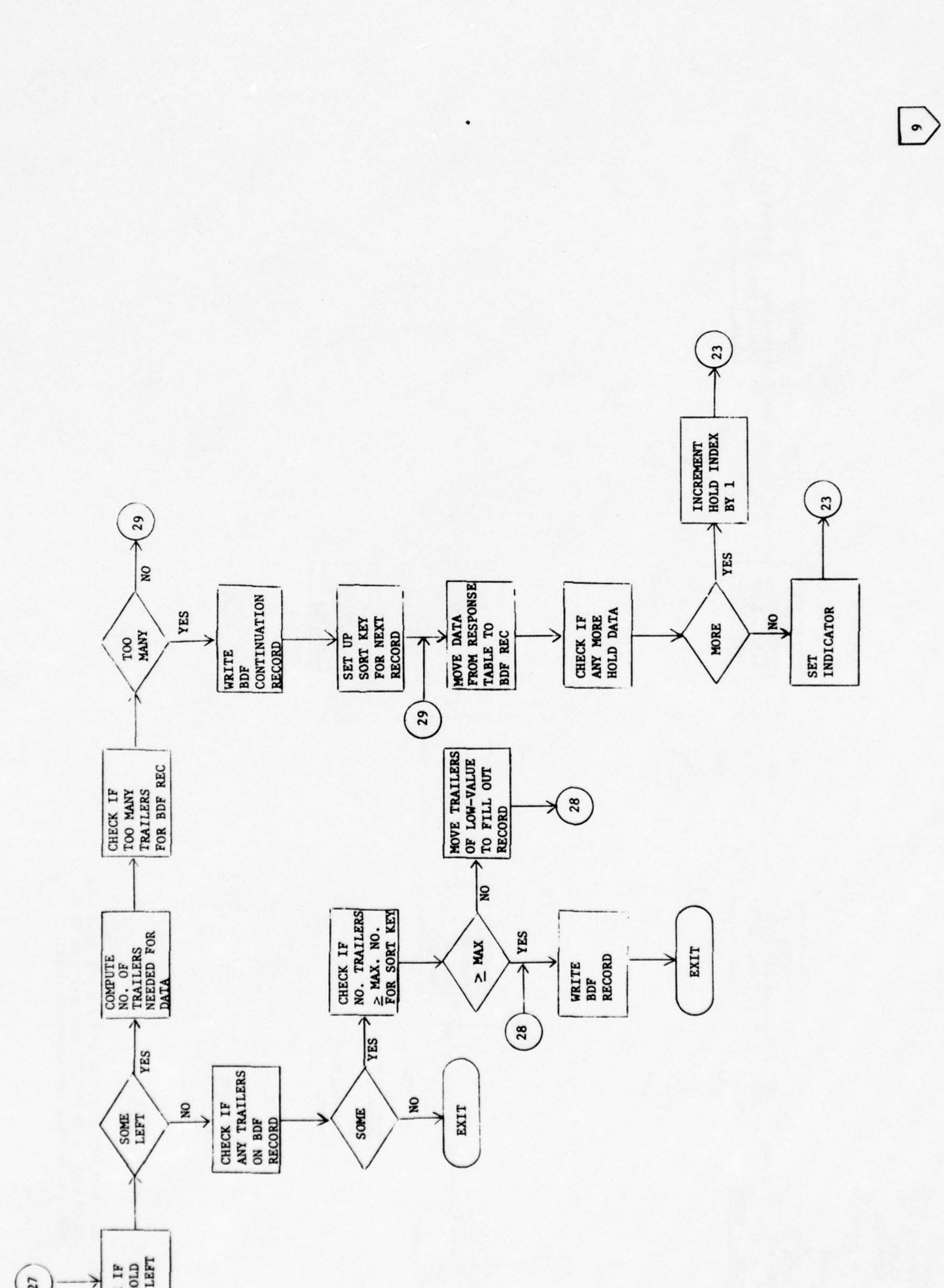


18
MOVE RESPONSE VALUE TO SORT KEY (CHAR.BY CHAR)
CHECK IF NO. OF CHARS MOVED ≥ 90
≥ 90
YES
NO
CHECK IF NO. OF CHARS. MOVED = STOP INDICATOR
EQUAL
YES
EXIT
NO
CHECK IF NEXT RESPONSE LIRC IS SAME
SAME
YES
NO
19
INCREMENT SORT RESPONSE TABLE INDEX BY 1
18
CHECK IF NO. OF CHARS MOVED = STOP INDICATOR
EQUAL
YES
EXIT
NO
19
7



*NOTE: HOLD DATA REFERS TO THE TABLE OF THE SORT LIRCS WHICH WERE SAVED BECAUSE THEY ARE ALSO TO BE USED FOR PRINTING AND/OR COMPUTATIONS.

8


27
CHECK IF ANY HOLD DATA LEFT
SOME LEFT
YES
NO
COMPUTE NO. OF TRAILERS NEEDED FOR DATA
CHECK IF TOO MANY TRAILERS FOR BDF REC
TOO MANY
YES
NO
29
WRITE BDF CONTINUATION RECORD
SET UP SORT KEY FOR NEXT RECORD
29
MOVE DATA FROM RESPONSE TABLE TO BDF REC
CHECK IF ANY MORE HOLD DATA
MORE
YES
NO
INCREMENT HOLD INDEX BY 1
23
SET INDICATOR
23
CHECK IF ANY TRAILERS ON BDF RECORD
SOME
YES
NO
EXIT
CHECK IF NO. TRAILERS ≥ MAX. NO. FOR SORT KEY
≥ MAX
YES
NO
MOVE TRAILERS OF LOW-VALUE TO FILL OUT RECORD
28
28
WRITE BDF RECORD
EXIT
9

A.9 PROGRAM TITLE: IMS Interface Module - DLIFSIF also called IMSIF

A.9.1 PURPOSE

This module is designed to be link edited with an applications program which updates an Information Management System (IMS) data base. As calls are made to IMS to update segments in the data base, this module will trap the calls made by the applications program and create a log of all updates to fields stored in NAVLIS inverted files. The module is to be used with both batch applications programs as well as message programs performing updates on-line. The log created can be read by the program IMSFOR (Section A.10) to create transaction records compatible with the SHARP inverted file update program.

A.9.2 INPUTS

Inputs to this module consist entirely of standard interface data available in the applications program. Specifically, parameters of the call from the applications program to IMS are interrogated after the call has been processed by IMS. The parameters used are:

- the argument count if provided
- the call function
- the applicable PCB in IMS
- the segment I-0 area in the applications program
- all Sesmont Search Arguments (SSAs) provided in the call

An additional input to this module is extracted on the first call. To create the update log, a Program Communication Block (PCB) must be provided in the applications program's Program Specification Block (PSB) generation. This PCB is located by examining the save area given to the application program by IMS. It is assumed that the application program saved register 1 in this save area. Using the address in register 1, the list of PCBs passed to the applications program is located. This list is scanned for either a PCB specifying the output

message queue (used for message programs) or the update log data base. The PCB names are respectively DLMSGO and DLTRNLOG.

Detail specifications of the structure of data used by the applications program can be found in the IMS Applications Programmers Guide and will not be discussed here. The present module provides an interface for COBOL and Assembler programs. No interface for PL/I programs was provided.

A.9.3 OUTPUTS

The output of this module can come in two forms depending on which output PCB was located. Since telprocessing PCBs must appear first in all PSB generations, the message queue PCB will be located first and that mode will be used in preference to the batch data base. The only difference in format between the on-line and batch modes is that the output generated is inherently variable length while the segments of a data base are fixed length. For this reason, a variable length segment is first generated within the module. In the on-line mode this segment can be directly inserted into the message queue. In the batch mode, this variable length segment is broken into 56-byte units and written, complete with length code, onto the data base until all required data have been placed in the data base.

The IMS access method used for the data base is the Hierarchical Direct Access Method (HDAM) although no randomizing is actually done. When the data base is initialized, a single root is placed in the data base by program IMSLOAD. Data are written onto the data base by inserting a dependent segment as a child to this single root. The segment inserted has no key and, as specified in the Data Base Description (DBD) generation will always be inserted after segments previously inserted to the data base. In effect, a sequential file has been created, except that full backup and recovery facilities of IMS are provided and subsequent executions of data base update programs will never over-write transactions already logged in this data base.

A.9.4 RECORD FORMATS

There are no applicable input record formats. The variable length output record is formatted as shown in Figure A17.

Byte	Format	Data Field
1-2	Halfword (H)	Record length
3-4	H	Binary zeros
5	Character (C)	Transaction mode. I = insert D = delete R = replace
6-8	C	NAVLIS file identifier. Always N01.
9-10	H	Segment identifier. Always H'1'.
11-var	C	Segment key. Variable length in general. Now always 13-byte key from WKACROOT.

For each field the following data are recorded. Each affected field is concatenated to form the rest of the record.

Format	Data Field
H	Length of this field. This is the number of bytes in the field itself not including this header data.
H	NAVLIS LIRC identifier for this field.
H	NAVLIS line count for this field if known. Binary zeros if no line count or unknown.
C	Value of the field. In replaces, this is the value before the replace was done.
C	Value of the field after a replace was done. This is present only for replaces and is the same length as the first value.

This record is inserted directly into the message queue if applicable. Inserts to the batch data base are made in 56-byte increments.

A.9.5 STORAGE REQUIREMENTS

For the Aircraft Engines Accounting data base of NMCSA (360/65) total storage required is 2910 bytes. In general, this is somewhat variable depending on the number of data bases, segments, and fields specified when the module is generated.

Section & Page 101.
Subsection
Date Documented
Change Notice #

Fold back at dotted line.

CARD OR RECORD LAYOUT – DOUBLE

SYSTEM ID:

☐ New ☐ Revision

RECORD TITLE:
FILE ID: IMSFOR INPUT FILE
RECORD LENGTH: VARIABLE (DEFINED IN IMSIF MACRO) MINIMUM ☐ FIXED ☐ MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR 1 RECORD/BLOCK

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON–CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7–TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Positions 1–2	3	4	5–6	7–8	9
REC LENGTH	IMS (1)	TRANS CODE (2)	FILE ID	SEGMENT ID	SEGMENT KEY (3)

FIELD ENTRIES: (4)
FIELD DATA LENGTH MAXSIZE = 2
LIRC OF FIELD MAXSIZE = 2
LINE COUNT MAXSIZE = 2
VALUE BEFORE REPLACE CALLS VARIABLE LENGTH
VALUE AFTER REPLACE CALLS VARIABLE LENGTH

(1) RESERVED FOR IMS
(2) "I" "D" OR "R"
(3) VARIABLE LENGTH
(4) VARIABLE LENGTH

RECORD TITLE:
FILE ID: IMSFOR OUTPUT FILE 1
RECORD LENGTH: 80 MINIMUM ☐ FIXED ☑ MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR 1 RECORD/BLOCK

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON–CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7–TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

Positions 1–2	3–6	7–10	11	12	13	14–23	24–36	37–39	40–89
FILE ID	FILLER	LIRC (1)	FILLER	ACTION CODE (2)	FILLER	HASHED KEY	FILLER	LINE CNT	FIELD VALUE

(1) NUMERIC
(2) "I" OR SPACE

Figure A-17

A.9.6 PROCESSING DESCRIPTION

Processing of an applications program begins at entry point CBLTDLI in this module. This entry point replaces the CBLTDLI entry point in the IMS module DFSLI000. To provide for expansion to languages other than COBOL, logic relating to interface with the applications program is separated from the call processing performed by this module. The language interface at CBLTDLI is responsible for passing the application program's call to IMS unchanged before branching to the actual processing routine at DLIMSIF. Prior to entry to DLIMSIF, a pointer to the applications program's call parameter list and the return address are placed in the normal linkage registers.

After IMS has completed the requested function, actual processing begins here at DLIMSIF. The first operation performed is to make sure that initialization operations are completed. On the first call, INITRTN is called to locate the PCB list provided by IMS and initialize appropriate tables in this module. When tables have been built, the next step is to determine whether the PCB supplied by the applications program is to be processed here. If the PCB does not refer to a data base whose updates are being trapped, no further processing is to be done. Control returns to the original caller. If, on the other hand, updates are to be trapped, the function code is tested and an appropriate routine is branched to. The routines are:

Function	Routine
All GH	GHRTN
ISRT	ISRTN
REPL	REPLRTN
DLET	DLETRTN
Other	Return to Caller

Processing is specific to the function specified. Routine GHRTN processes all Get-Hold (GH) calls. Its function is to store the "before" images for later comparison on updates. Notice that IMS requires the user to first issue a GH call before a REPL or DLET can be performed.

The GH routine must cope with:

(1) Invalid Status Indicators returned by IMS. This includes segments requested which do not exist in this data base.
(2) Calls using qualification on multiple levels within the data base using multiple SSAs.
(3) Calls requesting that segments from more than one level in the data base be concatenated in the user's I-0 area.

Status codes are tested first. Only if the code indicates that data could have been returned to the applications progr-m will processing continue here. The level code in the PCB will be used to determine exactly which segments were actually returned. After some initialization, processing proceeds SSA by SSA.

The segment entry is located in the tables generated for this database using the segment name specified in the SSA. If command codes in the SSA indicate that data were placed in the user's I-0 area for this segment, the fields of interest are extracted from the I-0 area and placed in a hold area in this module. Concatenations caused by path calls (command code D) can be processed by adjusting a pointer to the user's I-0 area as SSAs are processed. The processing of SSAs is terminated either by an SSA which is not found in this data base's table entry or by one whose level exceeds the maximum level returned to the user as indicated by the level code in the PCB.

Since similar operations of extracting selected fields from the users I-0 area are used for GET HOLD and REPLACE calls, a common routine MVSEGFIO is used both in GHRTN and ISRTN.

Processing of inserts is similar to that described for GET HOLD calls. The primary difference is that, as fields are extracted from the user's I-0 area, the data are placed directly in an output record area rather than in a hold area. A second difference is in the interpretation of command code D in the SSA. In a GET call, the D code is used for every segment to be retrieved. On an INSERT call, the D code indicates that this and all lower level segments in the list of SSA's are concatenated in the user's I-0 area.

Processing of REPLACE calls is done by REPLRTN. The first test for other calls is to see whether IMS returned a successful status code. If so, processing proceeds SSA by SSA. Each SSA is tested for an N command code indicating that data for the segment are present in the user's I-O area but are not to be processed by the replace operation. Whenever an N is found, the hold area is searched for an indication that data have been stored for the segment. Since no data were changed by the call in this case, no comparison is made here for changed fields for the segment named in the SSA. After this test is made, routine MVSEGFHA will be used to compare fields in the user's I-O area with those stored in the hold area by the previous GET HOLD call. Entries are placed in the output record for changed fields only.

Processing of deletes is done by routine DLETRTN. It is similar to REPLACE call processing. The first SSA provided by the caller indicates the lowest level segment to be deleted. Segments of a higher level stored in the hold area from the preceding GET HOLD call are not processed for deletes. Segments which were deleted are processed by routine MVSEGFHA which creates the output record reflecting the fields previously contained in the deleted segments.

Initialization is performed by routine INITRTN. The first task performed is the location of the original PCB list through save areas. It is assumed that the highest level save area belongs to the region controller. Hence the next save area in the chain contains registers as passed to the applications program. The highest level save area is located by an EXTRACT macro which returns the location of a field of PCB. A fixed off-set from this is the pointer to the highest level save area. Once the PCB list is located through the register 1 value, a scan is made of the entries. Each data base specified in a PCB is inspected to see whether it is to be processed by this routine. If it is, an entry is made in the PCB table here, and hold area space is allocated for storage of fields returned on GET HOLD calls. A test is also made for the output message queue or data base PCB's. The first one found is saved for later use in calls to actually insert the logged data through IMS.

Change data are logged by the routine PUTOUT. If the output message queue is being used, an INSERT call is made to IMS (using entry point DFSASM). If an output data base is used, 56-byte segments are inserted until all change data have been written.

MACROS

The following MACROS have been provided to generate the block describing segments and fields of interest to NAVLIS. Keyword parameters are also listed.

DEFDBD - Defines a DBD. This must be coded prior to other definitions.

- NAME - Name of DBD
- SEG1 - Primary segment for update log.
- SEG2 - Secondary segment for update log. This will be a dependent of SEG1.

DEFSEG - This must be coded prior to any field definitions. If a lower level segment is defined, a complete hierarchial sequence.

- NAME - Segment name - omit if a continuation.
- LEVEL - Level of segment. This must be two digits. Default is 01.
- FILE - NAVLIS file ID
- ID - NAVLIS Segment Identifier; default is 1.
- IOLEN - Length of I-O area occupied by this segment. Default is 0. This must be correct for processing of path calls. Consider omitting it on continuation definitions.
- KFBLEN - Length of key data to be taken from PCB key feedback area. Default is 0.
- KEYPOS - Starting position in I-O area from which key data are to be taken.
- KEYLEN - Length of key data in I-O area
- ISRT - Yes or No. Code No if inserts are not to be logged. Default is Yes.
- DLET - Yes or No. Code No if deletes are not to be logged. Default is Yes.

REPL - Yes or No. Code No if replaces are not to be logged. Default is Yes.

CONT - Yes or No. Code Yes if this segment defines the same segment as the one previously defined. This is used if new parameters are to be supplied for the fields defined for this segment, particularly File and ID in logical data bases.

DEFFLD - Defines a field

LIRC - NAVLIS LIRC for this field

LINECT - Line count of the field. Default is 0.

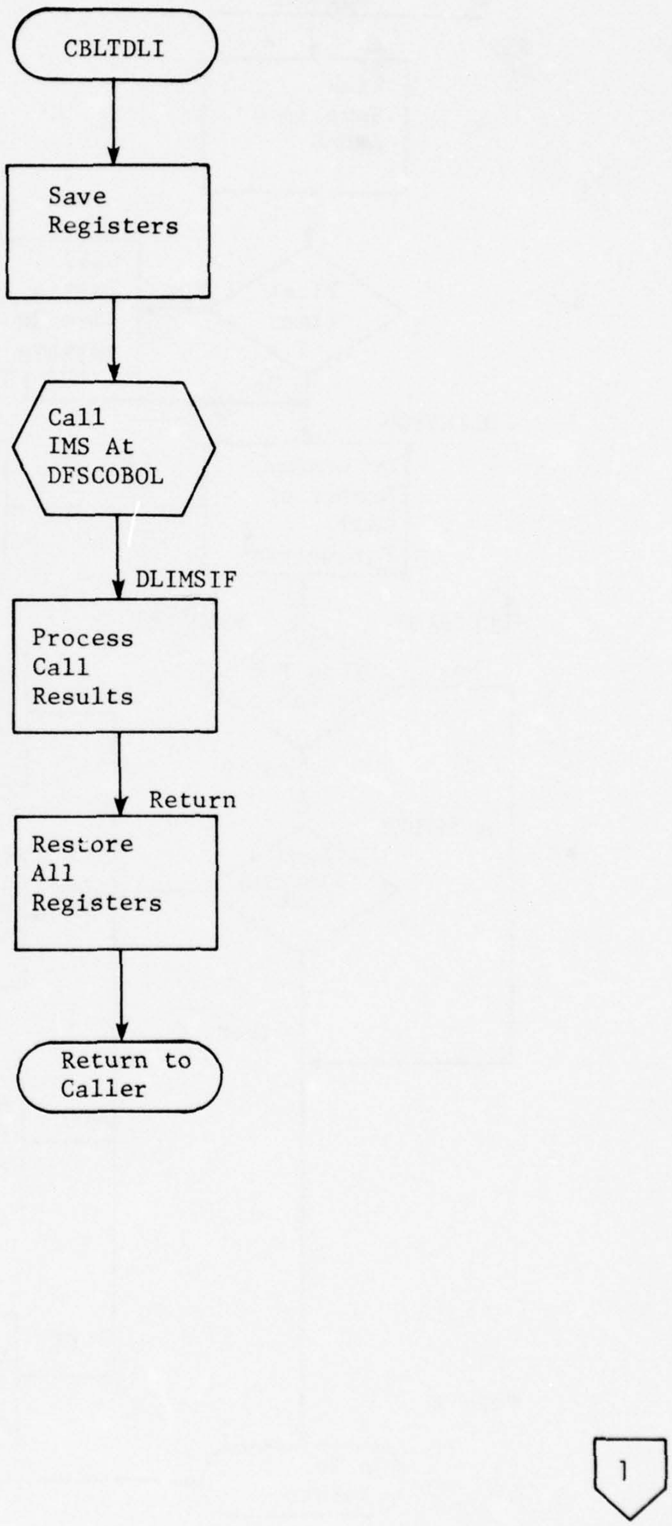
START - Starting position in I-O area.

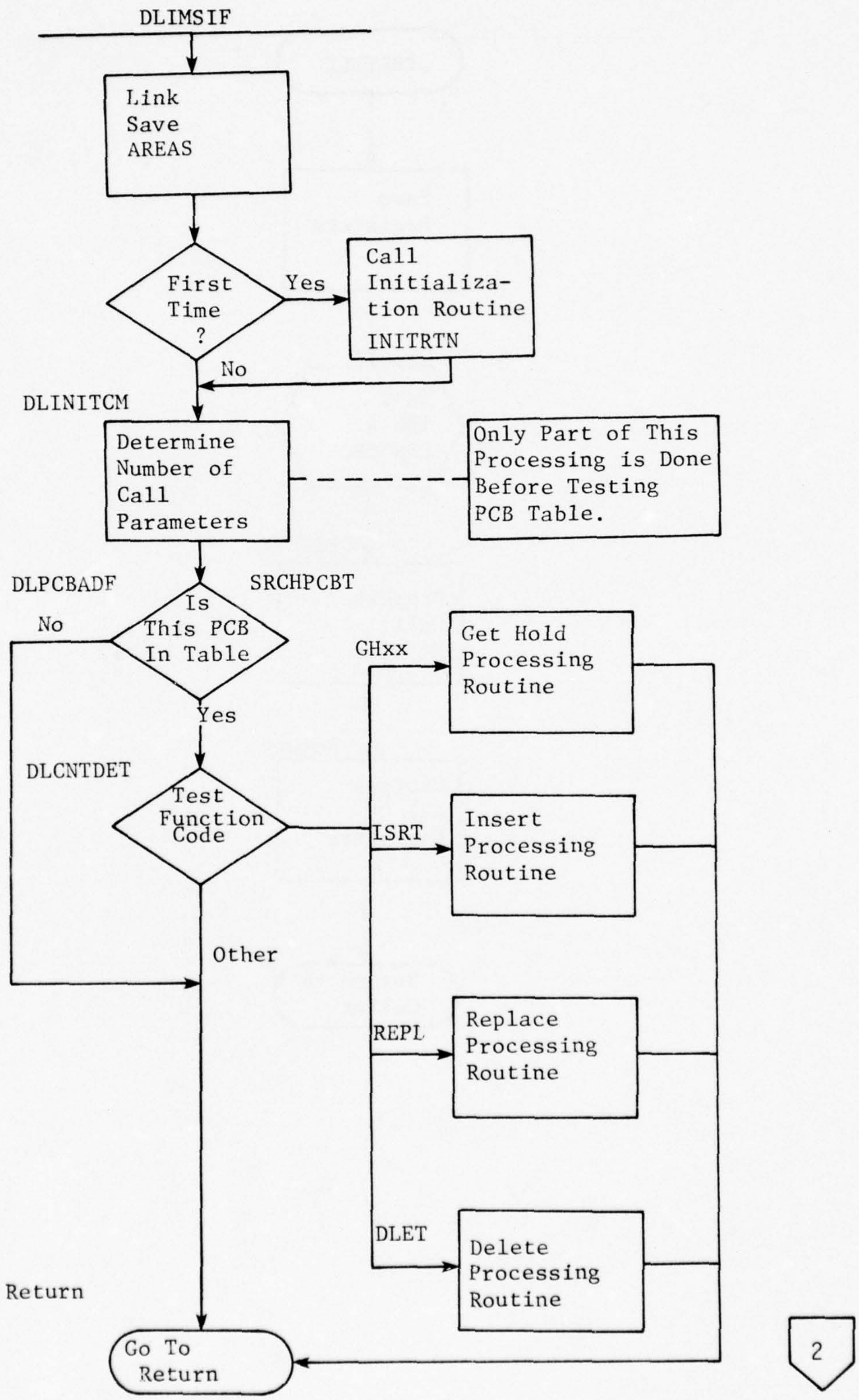
LENGTH - Length of this field.

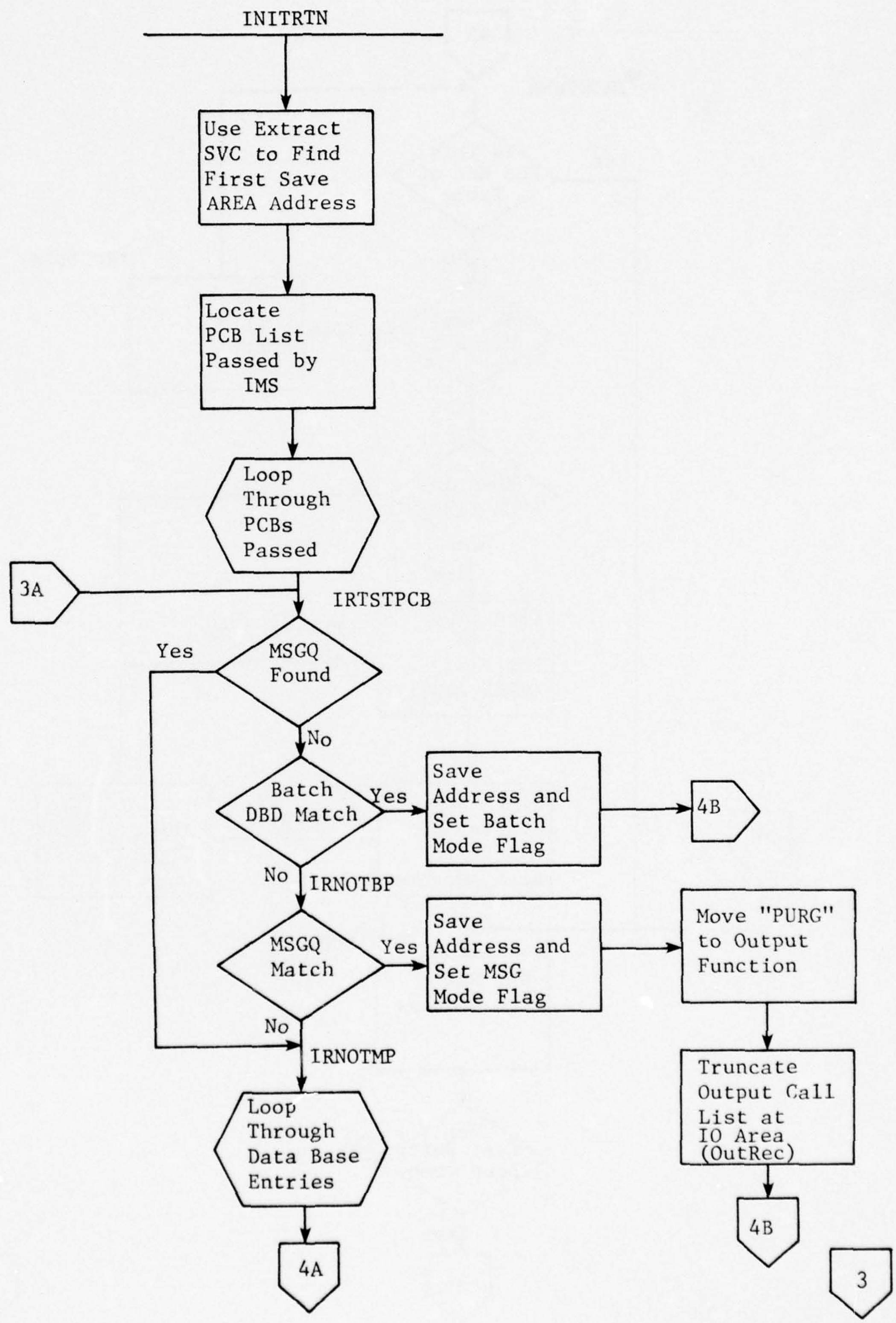
DBDEND - End of definitions. This must be the last MACRO coded.

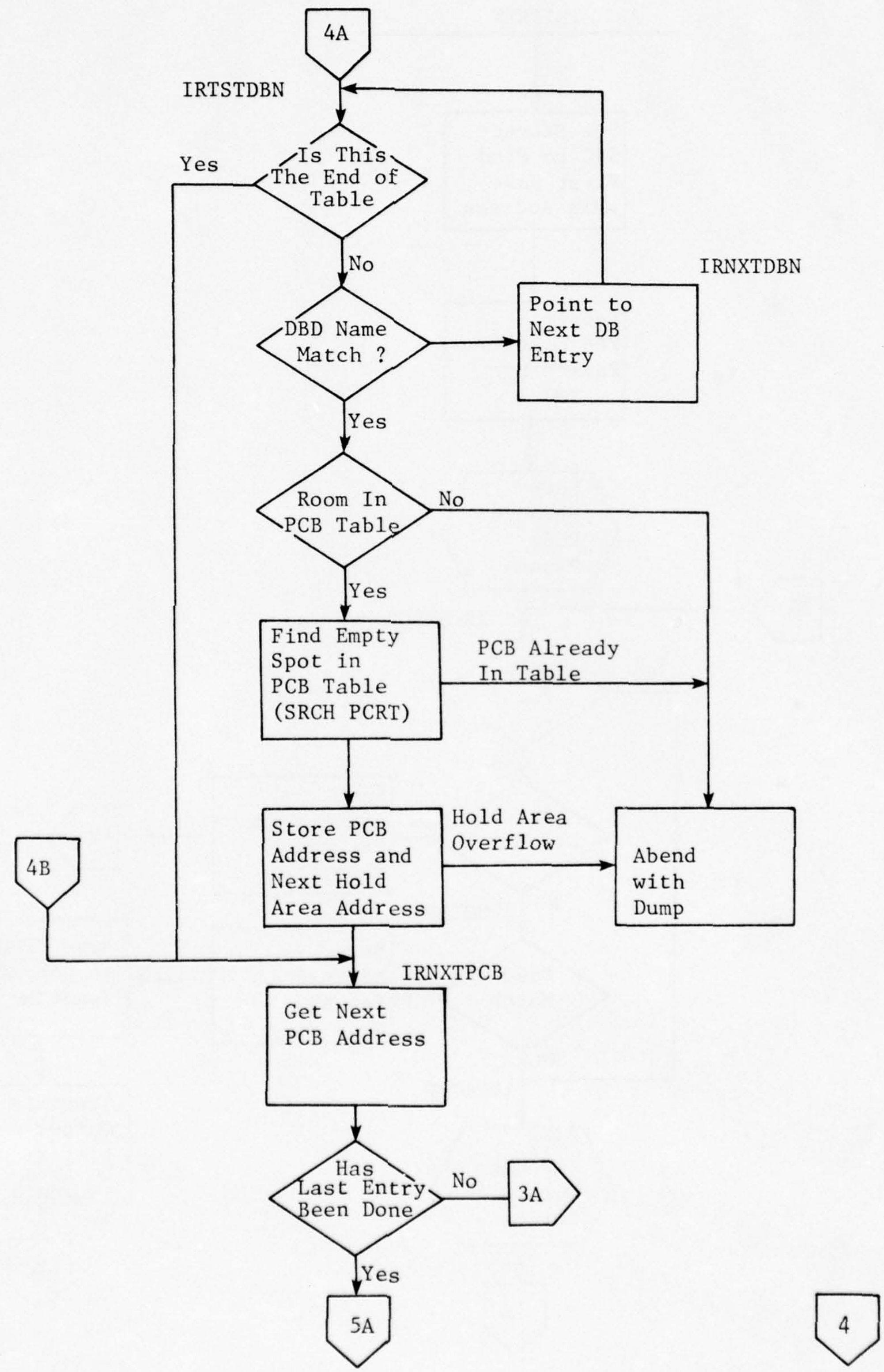
For generation of the interface module, a MACRO is provided:

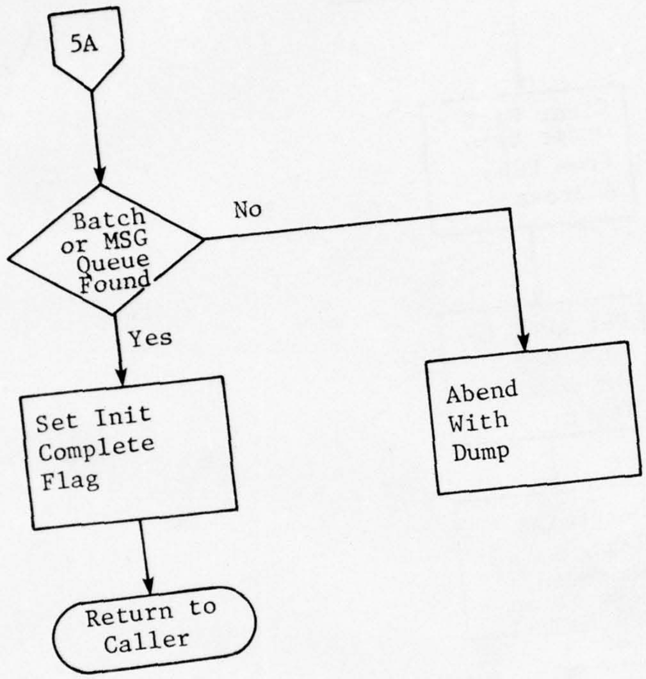
IMSIF - Generates interface routine.


CBLTDLI
Save Registers
Call IMS At DFSCOBOL
DLIMSIF
Process Call Results
Return
Restore All Registers
Return to Caller
1

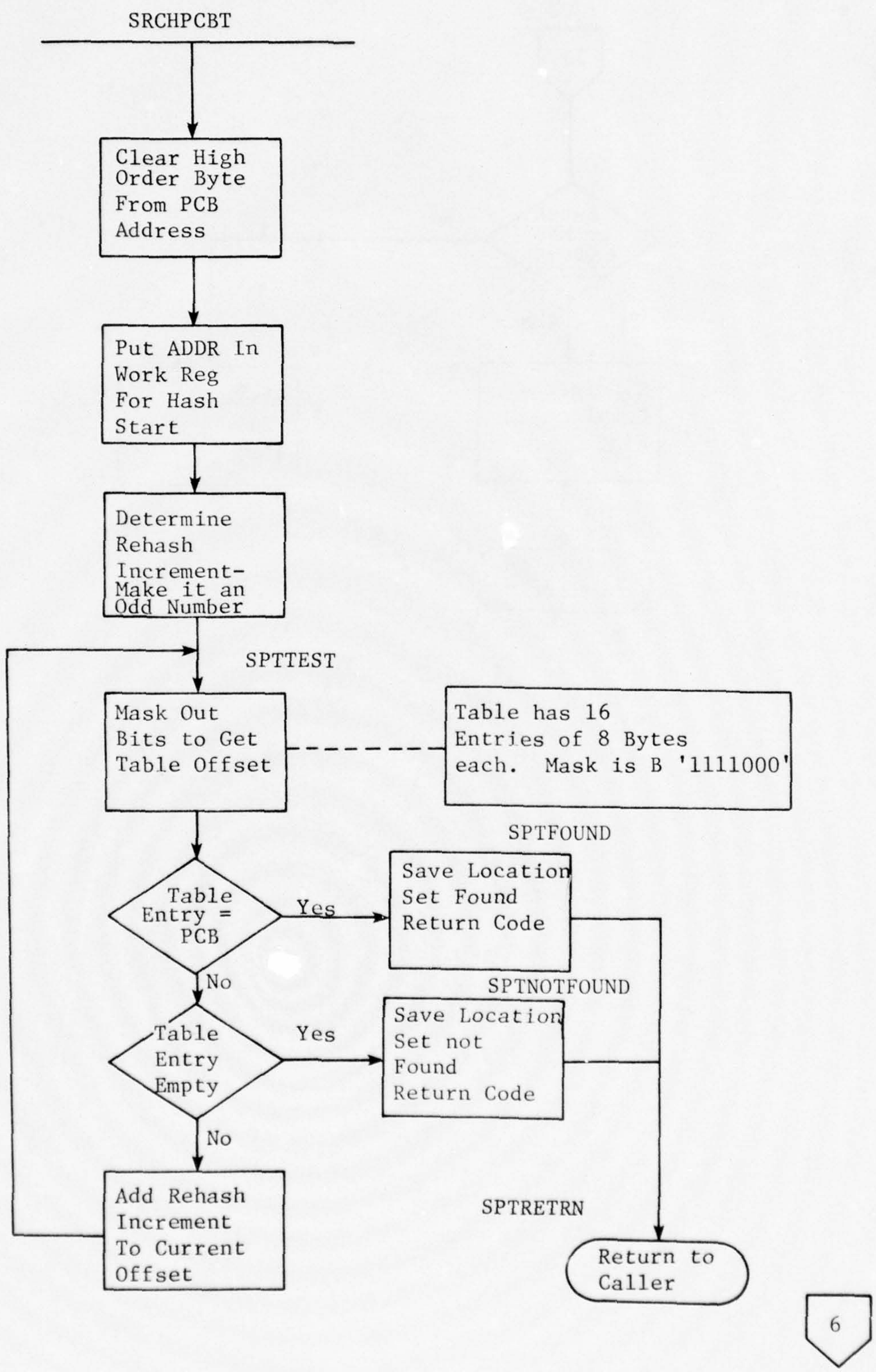

DLIMSIF
Link
Save
AREAS
First
Time
?
Yes
Call
Initializa-
tion Routine
INITRTN
No
DLINITCM
Determine
Number of
Call
Parameters
Only Part of This
Processing is Done
Before Testing
PCB Table.
DLPCBADF
SRCHPCBT
No
Is
This PCB
In Table
Yes
GHxx
Get Hold
Processing
Routine
DLCNTDET
Test
Function
Code
ISRT
Insert
Processing
Routine
Other
REPL
Replace
Processing
Routine
DLET
Delete
Processing
Routine
Return
Go To
Return
2

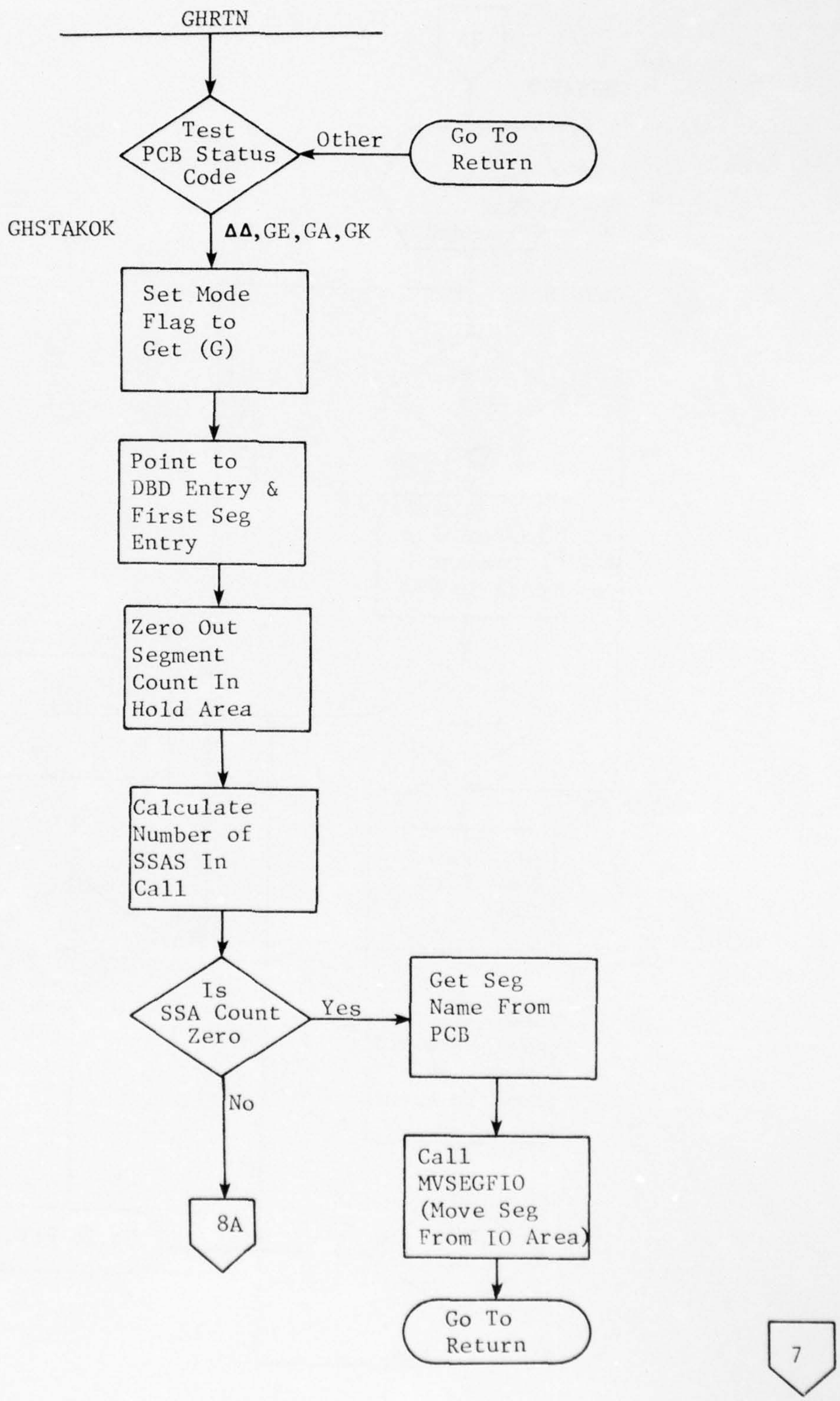

INITRTN
Use Extract SVC to Find First Save AREA Address
Locate PCB List Passed by IMS
Loop Through PCBs Passed
3A
IRTSTPCB
MSGQ Found
Yes
No
Batch DBD Match
Yes
Save Address and Set Batch Mode Flag
4B
No
IRNOTBP
MSGQ Match
Yes
Save Address and Set MSG Mode Flag
Move "PURG" to Output Function
Truncate Output Call List at IO Area (OutRec)
4B
No
IRNOTMP
Loop Through Data Base Entries
4A
3

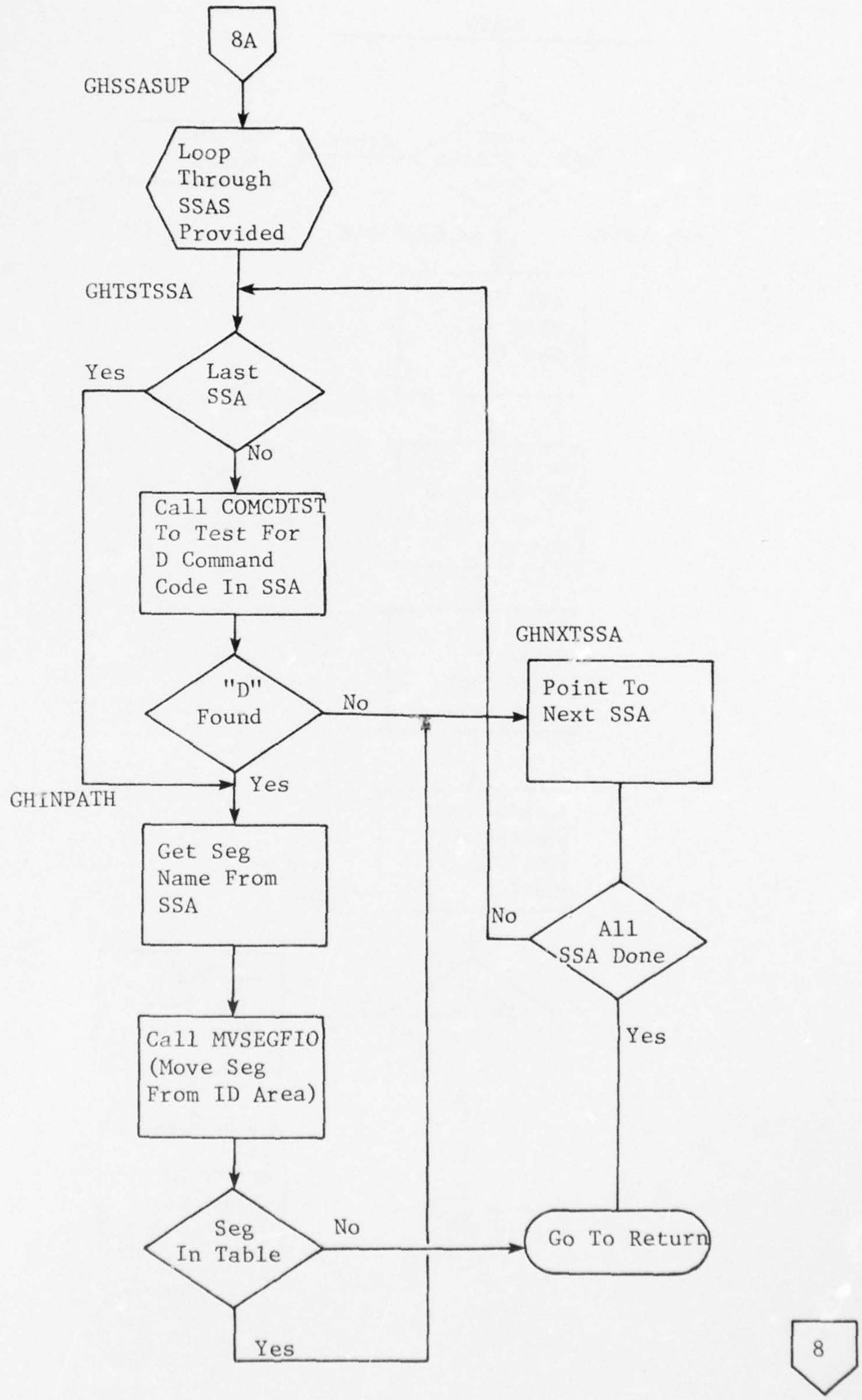

4A
IRTSTDBN
Is This The End of Table
Yes
No
DBD Name Match ?
IRNXTDBN
Point to Next DB Entry
Yes
Room In PCB Table
No
Yes
Find Empty Spot in PCB Table (SRCH PCRT)
PCB Already In Table
Store PCB Address and Next Hold Area Address
Hold Area Overflow
Abend with Dump
4B
IRNXTPCB
Get Next PCB Address
Has Last Entry Been Done
No
3A
Yes
5A
4

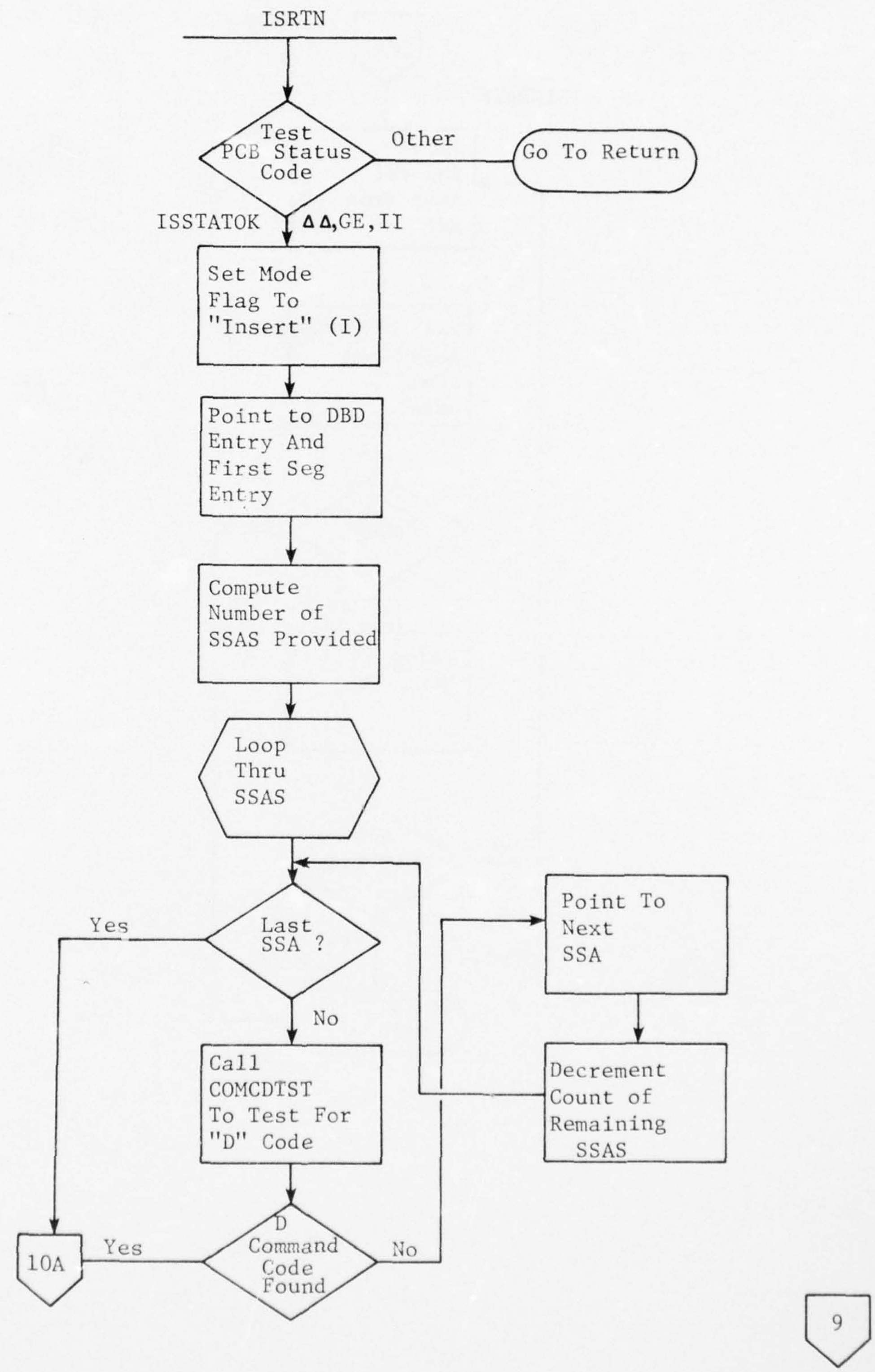

5A
Batch or MSG Queue Found
No
Yes
Set Init Complete Flag
Abend With Dump
Return to Caller

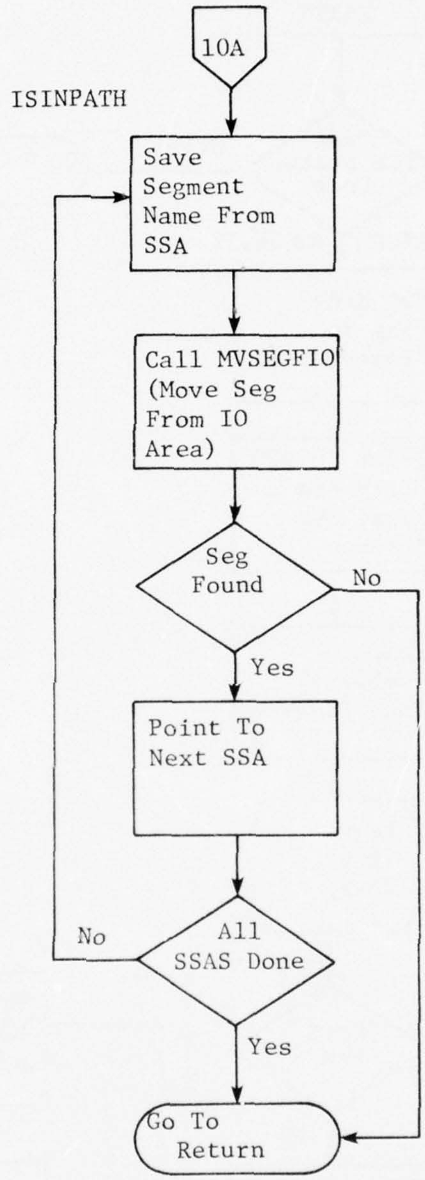

5


SRCHPCBT
Clear High Order Byte From PCB Address
Put ADDR In Work Reg For Hash Start
Determine Rehash Increment- Make it an Odd Number
SPTTEST
Mask Out Bits to Get Table Offset
Table has 16 Entries of 8 Bytes each. Mask is B '1111000'
SPTFOUND
Table Entry = PCB
Yes
Save Location Set Found Return Code
No
SPTNOTFOUND
Table Entry Empty
Yes
Save Location Set not Found Return Code
No
Add Rehash Increment To Current Offset
SPTRETRN
Return to Caller
6

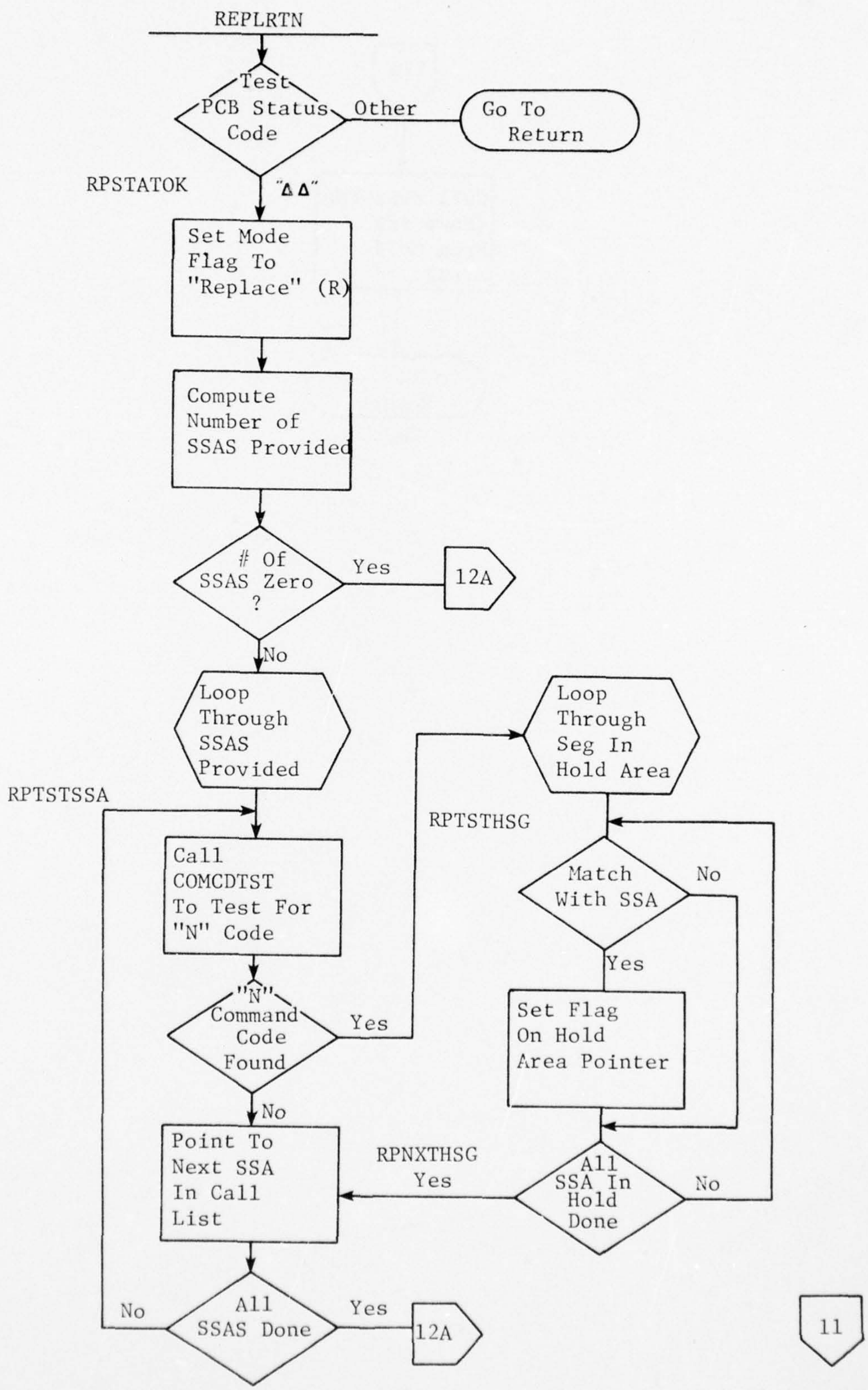

GHRTN
Test PCB Status Code
Other
Go To Return
GHSTAKOK
ΔΔ,GE,GA,GK
Set Mode Flag to Get (G)
Point to DBD Entry & First Seg Entry
Zero Out Segment Count In Hold Area
Calculate Number of SSAS In Call
Is SSA Count Zero
Yes
Get Seg Name From PCB
No
Call MVSEGFIO (Move Seg From IO Area)
8A
Go To Return
7

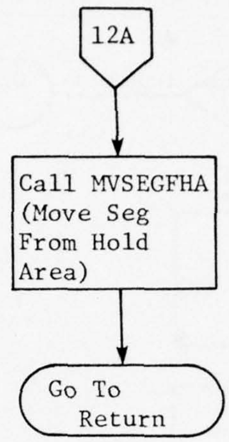

8A
GHSSASUP
Loop Through SSAS Provided
GHTSTSSA
Last SSA
Yes
No
Call COMCDTST To Test For D Command Code In SSA
"D" Found
No
Yes
GHINPATH
Get Seg Name From SSA
Call MVSEGFIO (Move Seg From ID Area)
Seg In Table
No
Yes
GHNXTSSA
Point To Next SSA
All SSA Done
No
Yes
Go To Return
8


ISRTN
Test PCB Status Code
Other
Go To Return
ISSTATOK
ΔΔ,GE,II
Set Mode Flag To "Insert" (I)
Point to DBD Entry And First Seg Entry
Compute Number of SSAS Provided
Loop Thru SSAS
Yes
Last SSA ?
No
Point To Next SSA
Decrement Count of Remaining SSAS
Call COMCDTST To Test For "D" Code
10A
Yes
D Command Code Found
No
9

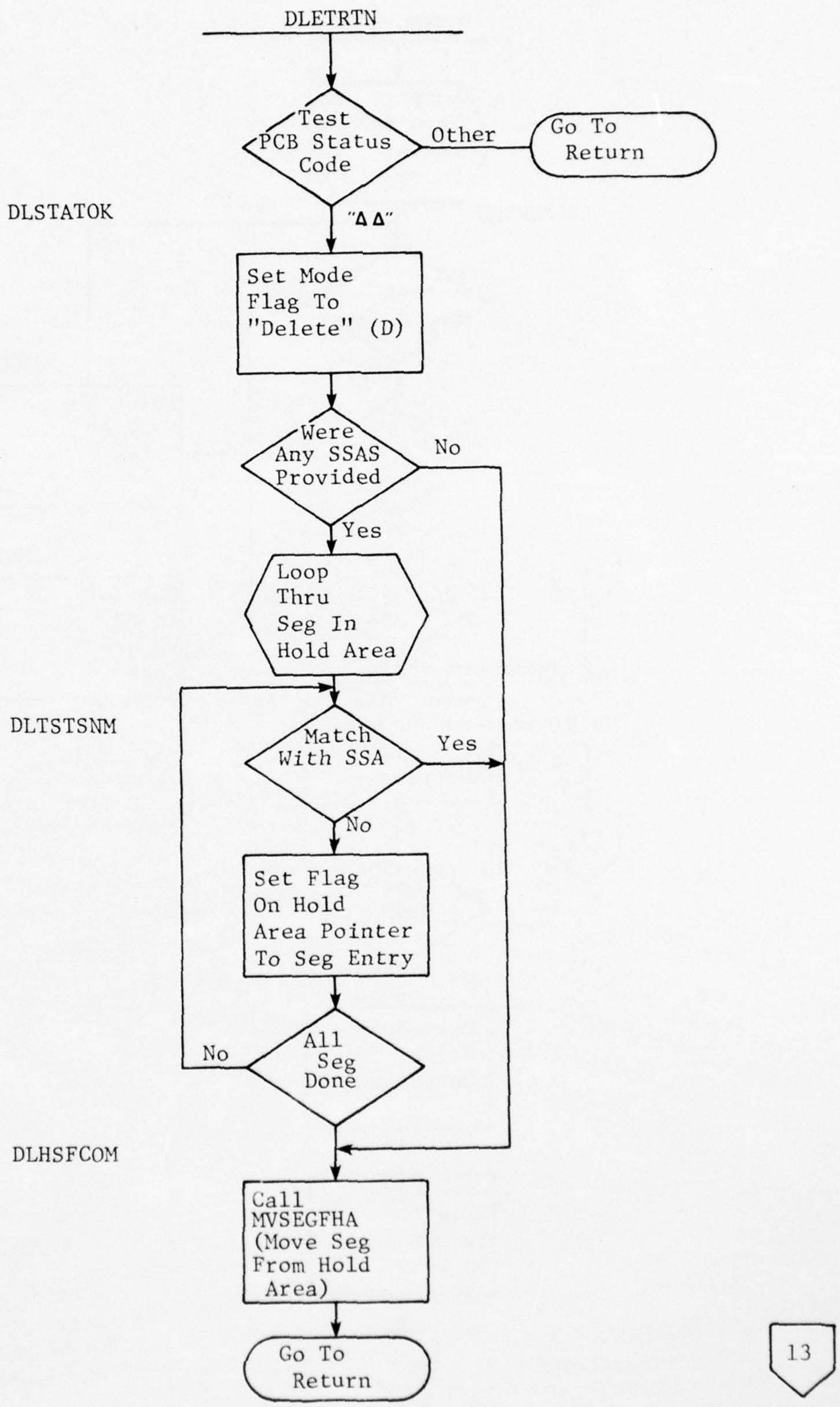

10A
ISINPATH
Save Segment Name From SSA
Call MVSEGFIO (Move Seg From IO Area)
Seg Found
No
Yes
Point To Next SSA
All SSAS Done
No
Yes
Go To Return

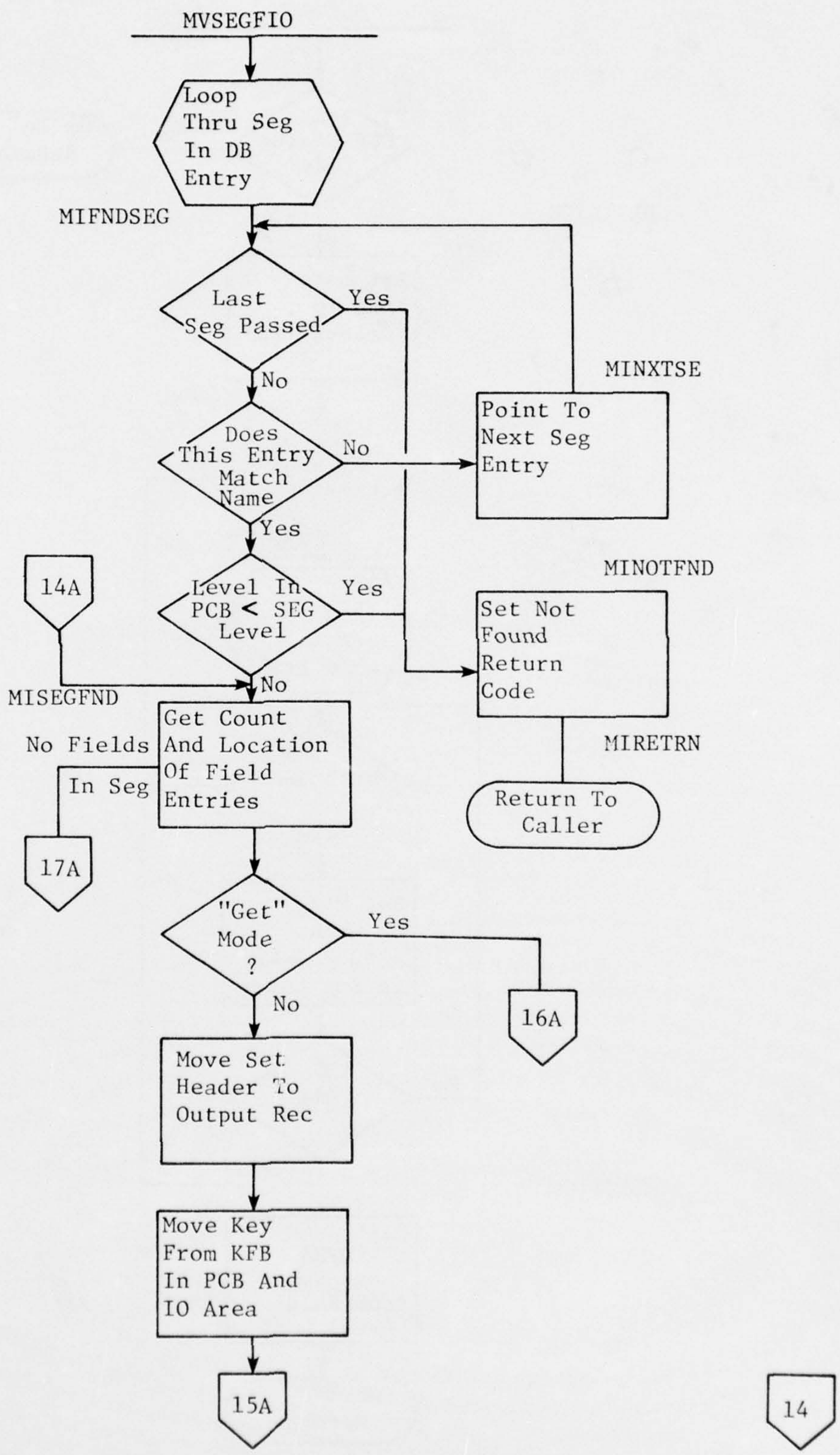
10

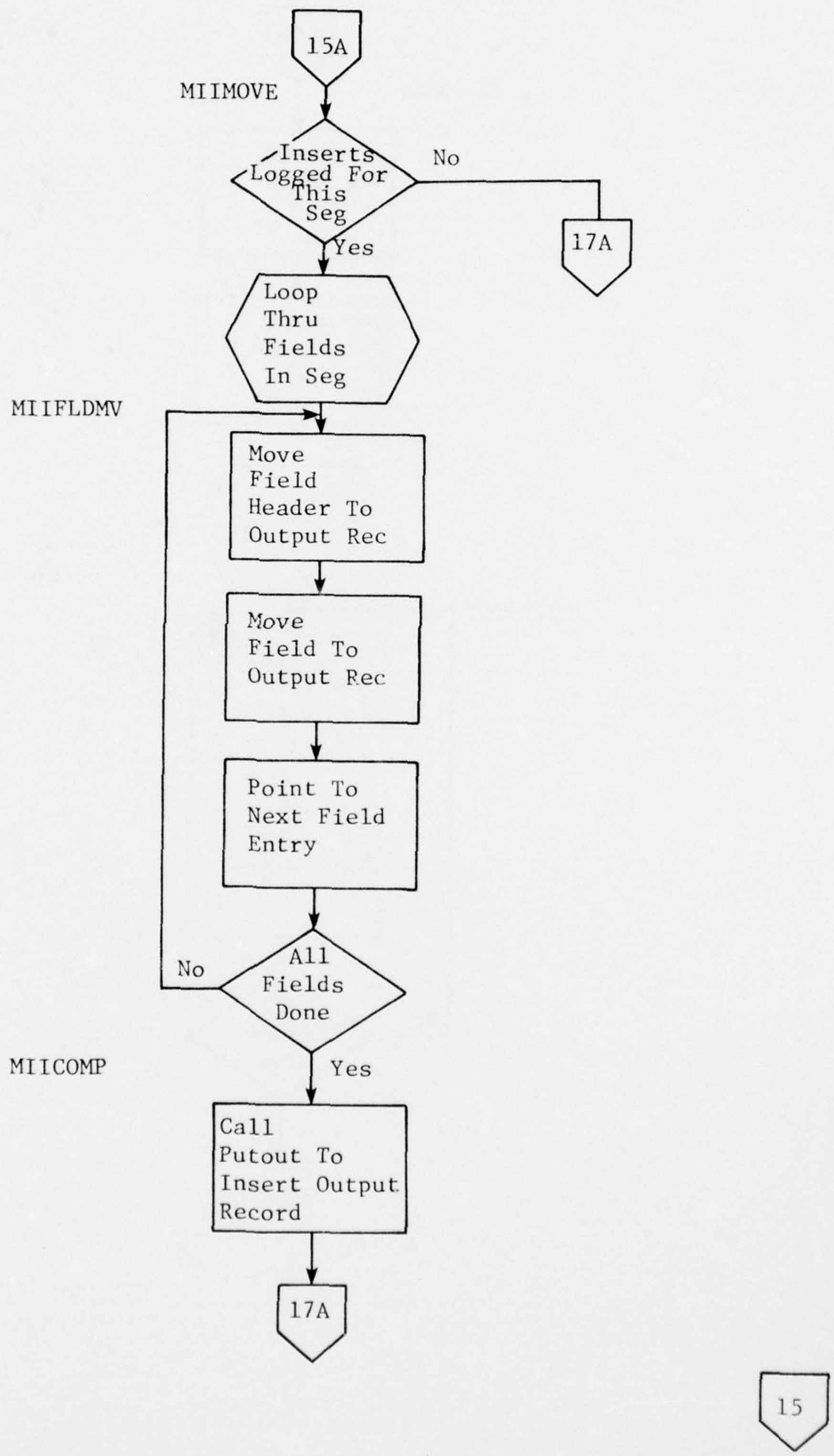

REPLRTN
Test PCB Status Code
Other
Go To Return
RPSTATOK
"ΔΔ"
Set Mode Flag To "Replace" (R)
Compute Number of SSAS Provided
Of SSAS Zero ?
Yes
12A
No
Loop Through SSAS Provided
RPTSTSSA
Call COMCDTST To Test For "N" Code
"N" Command Code Found
Yes
No
Loop Through Seg In Hold Area
RPTSTHSG
Match With SSA
No
Yes
Set Flag On Hold Area Pointer
Point To Next SSA In Call List
RPNXTHSG
Yes
All SSA In Hold Done
No
No
All SSAS Done
Yes
12A
11

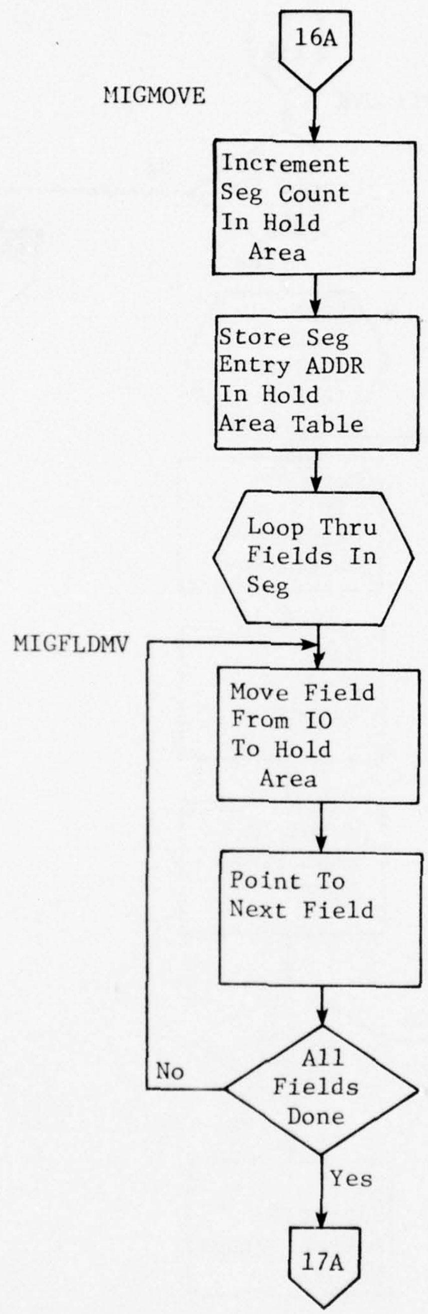

12A
Call MVSEGFHA (Move Seg From Hold Area)
Go To Return

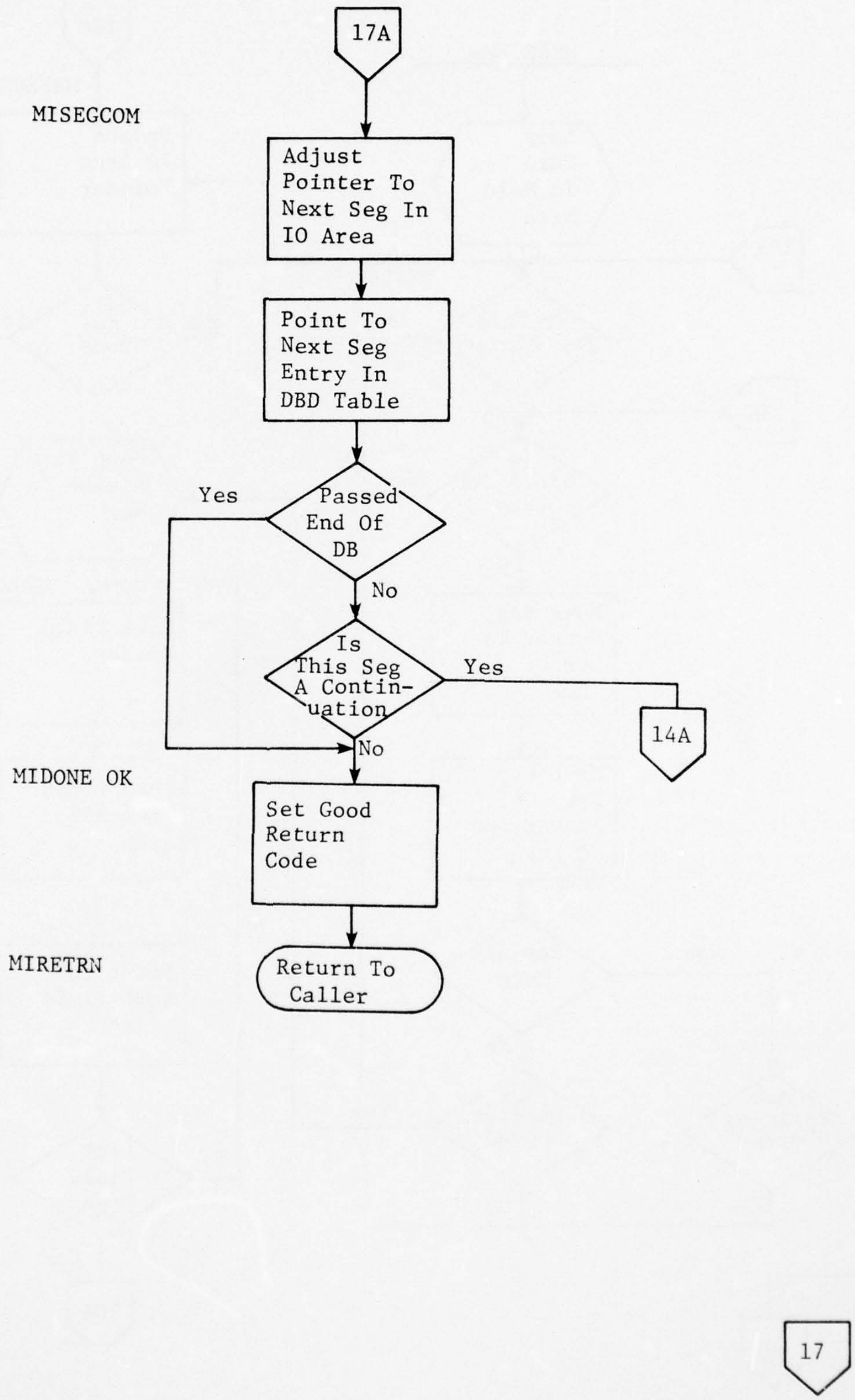

12

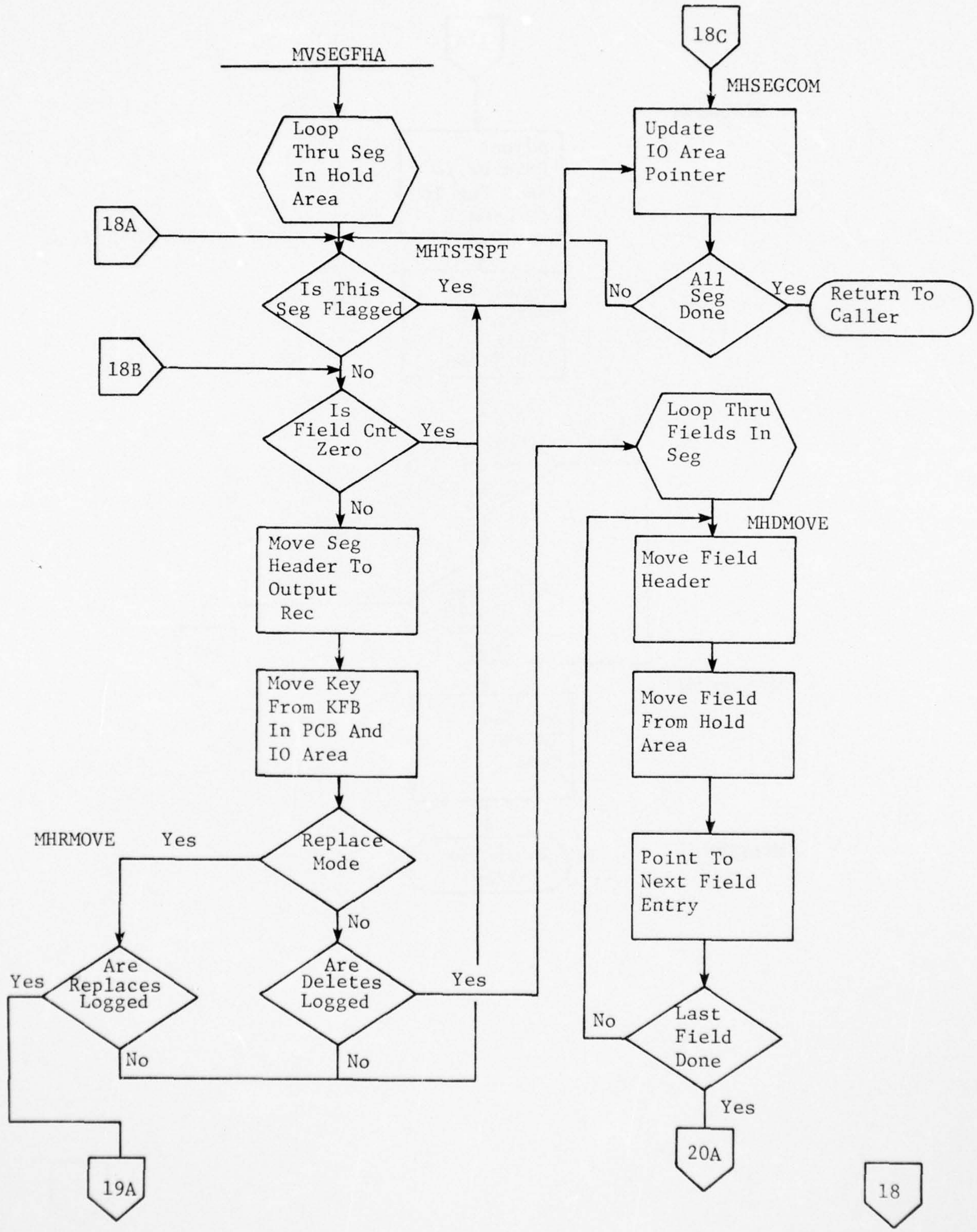

DLETRTN
Test PCB Status Code
Other
Go To Return
DLSTATOK
"ΔΔ"
Set Mode Flag To "Delete" (D)
Were Any SSAS Provided
No
Yes
Loop Thru Seg In Hold Area
DLTSTSNM
Match With SSA
Yes
No
Set Flag On Hold Area Pointer To Seg Entry
All Seg Done
No
DLHSFCOM
Call MVSEGFHA (Move Seg From Hold Area)
Go To Return
13

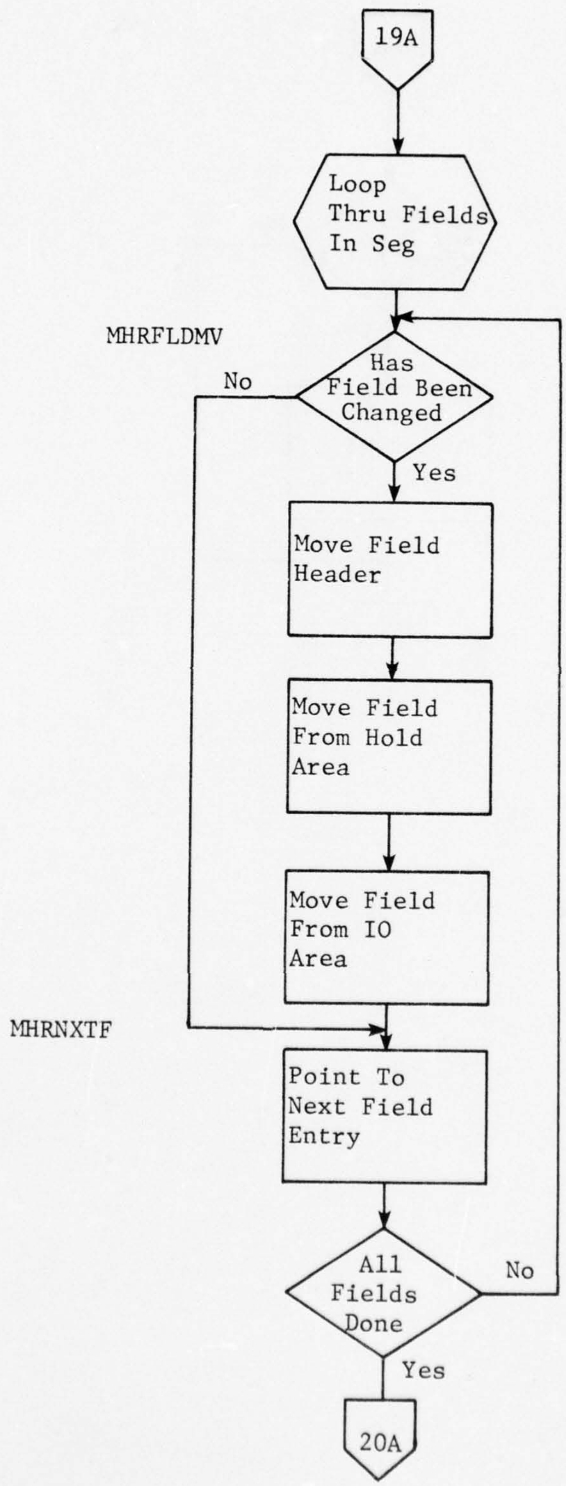

MVSEGFIO
Loop Thru Seg In DB Entry
MIFNDSEG
Last Seg Passed
Yes
No
MINXTSE
Point To Next Seg Entry
Does This Entry Match Name
No
Yes
14A
Level In PCB < SEG Level
Yes
MINOTFND
Set Not Found Return Code
No
MISEGFND
Get Count And Location Of Field Entries
No Fields In Seg
MIRETRN
Return To Caller
17A
"Get" Mode ?
Yes
No
16A
Move Set Header To Output Rec
Move Key From KFB In PCB And IO Area
15A
14


15A
MIIMOVE
Inserts Logged For This Seg
No
17A
Yes
Loop Thru Fields In Seg
MIIFLDMV
Move Field Header To Output Rec
Move Field To Output Rec
Point To Next Field Entry
No
All Fields Done
MIICOMP
Yes
Call Putout To Insert Output Record
17A
15

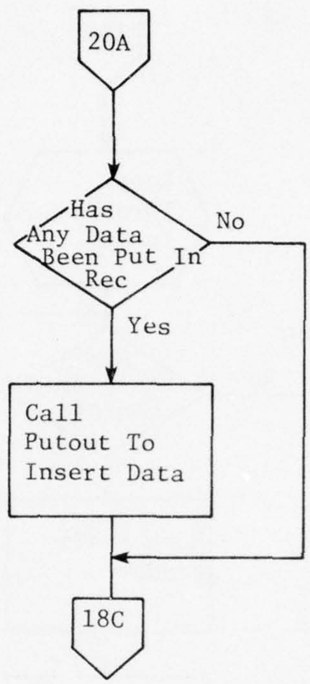

16A
MIGMOVE
Increment Seg Count In Hold Area
Store Seg Entry ADDR In Hold Area Table
Loop Thru Fields In Seg
MIGFLDMV
Move Field From IO To Hold Area
Point To Next Field
All Fields Done
No
Yes
17A


17A
MISEGCOM
Adjust Pointer To Next Seg In IO Area
Point To Next Seg Entry In DBD Table
Passed End Of DB
Yes
No
Is This Seg A Contin-uation
Yes
14A
No
MIDONE OK
Set Good Return Code
MIRETRN
Return To Caller
17

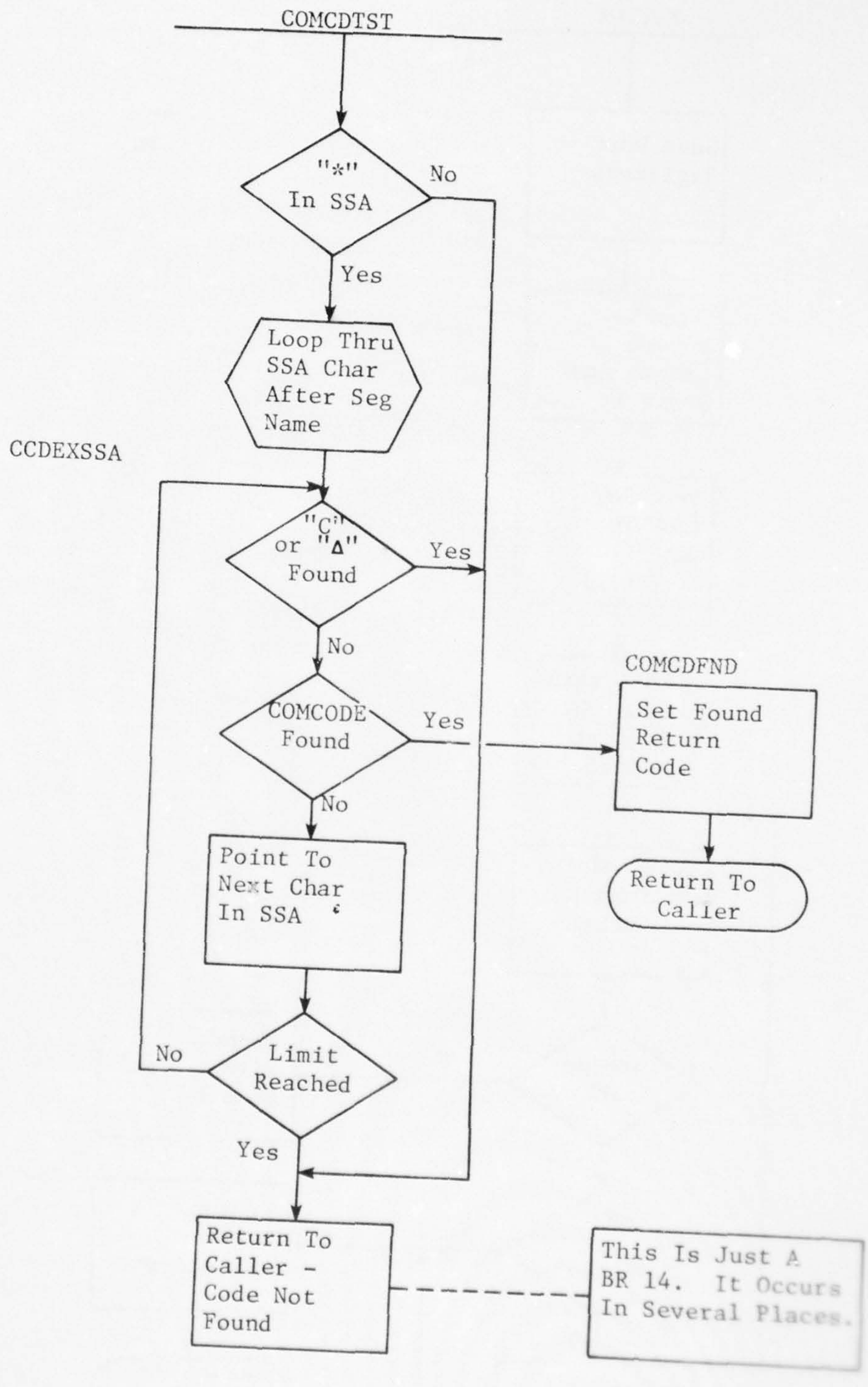

MVSEGFHA
Loop Thru Seg In Hold Area
18A
MHTSTSPT
Is This Seg Flagged
Yes
18B
No
Is Field Cnt Zero
Yes
No
Move Seg Header To Output Rec
Move Key From KFB In PCB And IO Area
MHRMOVE
Yes
Replace Mode
No
Yes
Are Replaces Logged
Are Deletes Logged
Yes
No
No
19A
18C
MHSEGCOM
Update IO Area Pointer
No
All Seg Done
Yes
Return To Caller
Loop Thru Fields In Seg
MHDMOVE
Move Field Header
Move Field From Hold Area
Point To Next Field Entry
No
Last Field Done
Yes
20A
18


19A
Loop
Thru Fields
In Seg
MHRFLDMV
Has
Field Been
Changed
No
Yes
Move Field
Header
Move Field
From Hold
Area
Move Field
From IO
Area
MHRNXTF
Point To
Next Field
Entry
All
Fields
Done
No
Yes
20A

19


20A
Has Any Data Been Put In Rec
No
Yes
Call Putout To Insert Data
18C


20


COMCDTST
"*"
In SSA
No
Yes
Loop Thru
SSA Char
After Seg
Name
CCDEXSSA
"C"
or "Δ"
Found
Yes
No
COMCODE
Found
Yes
No
COMCDFND
Set Found
Return
Code
Return To
Caller
Point To
Next Char
In SSA
No
Limit
Reached
Yes
Return To
Caller -
Code Not
Found
This Is Just A
BR 14. It Occurs
In Several Places.

AD-A035 847 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 9/2

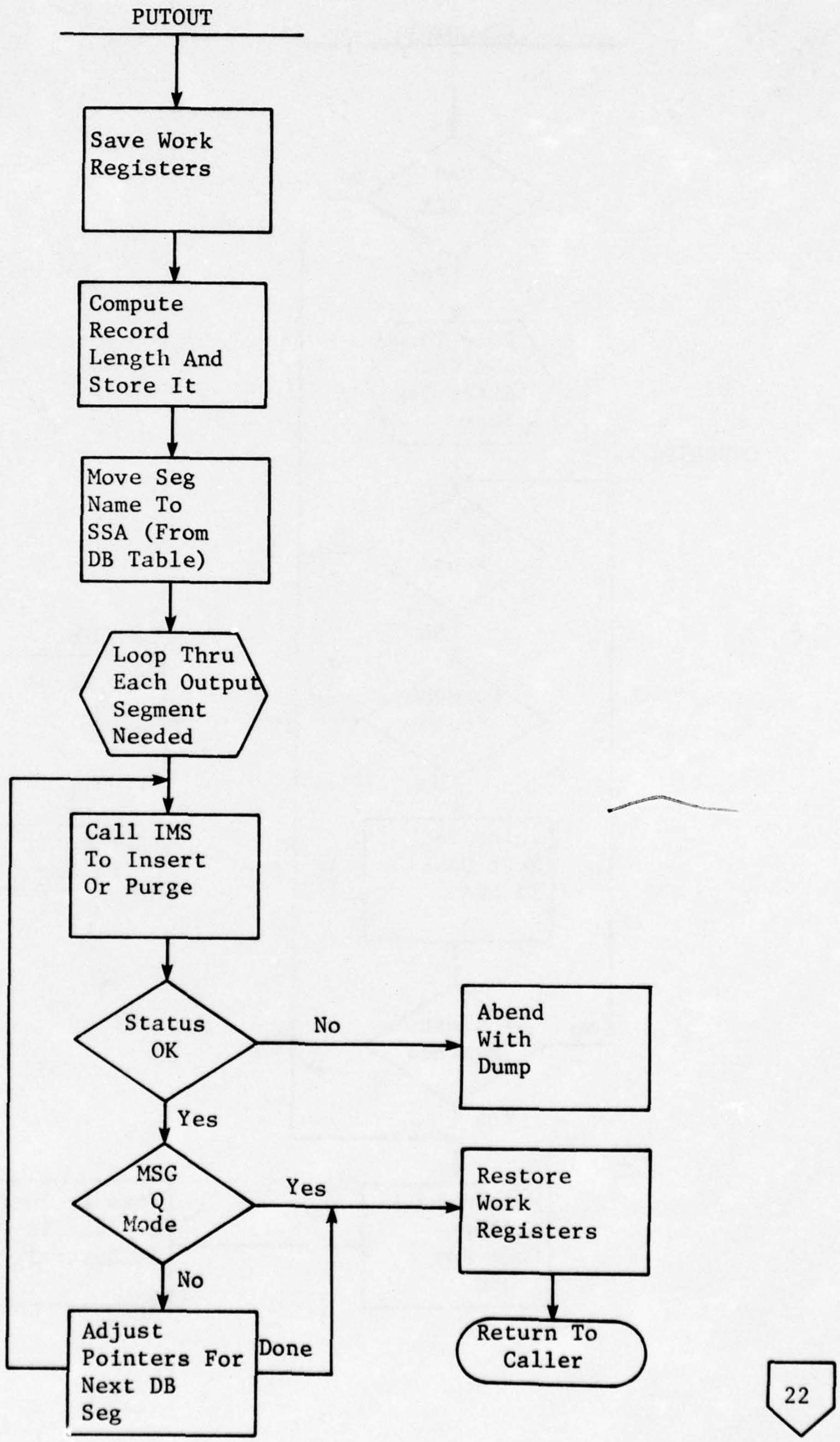
NAVY LOGISTICS INFORMATION SHARING (NAVLIS) PROJECT.(U)

JUN 76

UNCLASSIFIED DTNSRDC-76-0120 NL

4 OF 5

AD A035847


PUTOUT
Save Work Registers
Compute Record Length And Store It
Move Seg Name To SSA (From DB Table)
Loop Thru Each Output Segment Needed
Call IMS To Insert Or Purge
Status OK
No
Abend With Dump
Yes
MSG Q Mode
Yes
Restore Work Registers
No
Adjust Pointers For Next DB Seg
Done
Return To Caller
22

A.10 TITLE; IMS FORMAT UPDATES MODULE - IMSFOR also called DLIMSFOR

A.10.1 PURPOSE

This module is used to convert the updates trapped by IMSIF to a format acceptable to the Inverted File Update Program. This module can be run in either a batch-message region or a batch-only region of IMS. In a batch-message region, input can be taken either from a message queue containing the updates or from a data base. Some operational impact may be noticed when using a data base as the input source since exclusive access to the data base is required.

A.10.2 INPUTS

Input to this module can be from two sources, either a data base or a message queue. The record formats are essentially identical. Segments from the message queue are formatted as follows:

Byte	Format	Data Field
1-2	H	Record length
3-4	H	Reserved for IMS
5	C	Transaction code; I = Insert, D = Delete, R = Replace
6-8	C	NAVLIS file identifier
9-10	H	Segment identifier
11-Var	C	Segment key

Remainder of the record is devoted to field entries. These entries are concatenated after the segment key. Each field entry consists of:

Byte	Format	Contents
1-2	H	Length of field data
3-4	H	LIRC of field
5-6	H	Line count of field if known, otherwise zero
7-Var	C	Value of field. In REPLACE calls this is value before replace attempted.
Var	C	Value of field - REPLACE calls only. Value after replace completed.

Segments in the data base are created by breaking this record into 56-byte units and inserting as many 56-byte segments as required to hold the data. The length code is used to tell how much data to process in the last segment.

A.10.3 OUTPUT

Output of this module is designed to be used by the standard SHARP Inverted File update program. The output here must be sorted to meet the requirements of SHARP. Records created to update the Inverted File are:

Byte	Format	Data Field
1-3	C	File ID - not used by SHARP
4-7	C	Not used
8-11	C	LIRC
12	C	Not used
13	C	Action Code Space = add to Inverted File 1 = delete from Inverted File
14	C	Not used
15-23	C	Hashed key of record
24-35	C	Not used
36-38	C	Line count
39-88	C	Value of the field

An additional function of this module is to add records to the Record Descriptor File for new records added to the data base. These records contain:

Byte	Format	Data Field
1	C	Delete byte - not used
2-10	C	Hash key of the data base record
11-40	C	Actual key of the data base record
41-46	C	Record number of data base record - not used

A.10.4 RECORD FORMATS: See Figure A18

A.10.5 STORAGE REQUIREMENTS

Program occupies approximately 11,000 bytes on IBM 360/65.

A.10.6 PROCESSING DESCRIPTION (Refer to IMSFOR Flowchart, page 220)

The first major decision made in this module is to determine the source of the input, i.e., message queue or data base. If input is from the data base, the initial root segment is retrieved since all transaction data are stored as dependent segments. Message queue retrieval does not require prior positioning.

Each input record is processed by retrieving a complete input record even if it spans multiple segments. If the action indicated is an INSERT, a new record must be added to the Record Description File; otherwise the existing record must be found so that the correct hash key can be determined. Both inserts and deletes require a transaction for the "0000" LIRC; these are "add" and "delete" respectively. All fields in the input record are then processed.

Processing of each field consists of moving File ID, LIRC, Line Count, Hash Key, and Field Value to the output record. Inserts have an action code of "add"; deletes and replaces have "delete". A second record is generated for a replace to add the new field value.

Processing used in adding new records to the Record Descriptor File and in locating existing ones is similar. In both cases, an initial hash key is determined using the same logic as INVERTAE.* Using the sequence of tie breaker characters that were used in building the file, it is searched until either the desired record is found or the record is found to be missing. It is assumed that no records are deleted - at least none are here. The expected result, when attempting to locate an existing record, is that the scan will terminate with the desired record. When a new record is to be added, the first missing hash key is assigned to the new record. If an error is detected, a message is displayed, corrective action taken, and processing continues.

* Program used to create the Inverted Master Files for the Aircraft Engines Accounting Data Base.

Section & Page 101.
Subsection
Date Documented
Change Notice #

← Fold back at dotted line.

CARD OR RECORD LAYOUT – DOUBLE

SYSTEM ID: ☐ New ☐ Revision

RECORD TITLE:
FILE ID: IMSFOR OUTPUT FILE 2
RECORD LENGTH: 46 MINIMUM ☐ FIXED ☑ MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☑
BLOCKING FACTOR 1 RECORD/BLOCK

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

HEADER:
TRAILER:
COLLATING SEQ:

SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

DELETE	HASHED KEY	ACTUAL KEY	RECORD NUMBER	
(1)			(2)	

(1) NOT USED
(2) NOT USED

RECORD TITLE:
FILE ID:
RECORD LENGTH: MINIMUM ☐ FIXED ☐ MAX.
WORDS ☐ BYTES ☐ CHARACTERS ☐
BLOCKING FACTOR

MEDIUM:
CARD ☐ TAPE ☐ DASD ☐
OTHER ☐
DENSITY
TRACKS: 7 ☐ 9 ☐
CARD STOCK

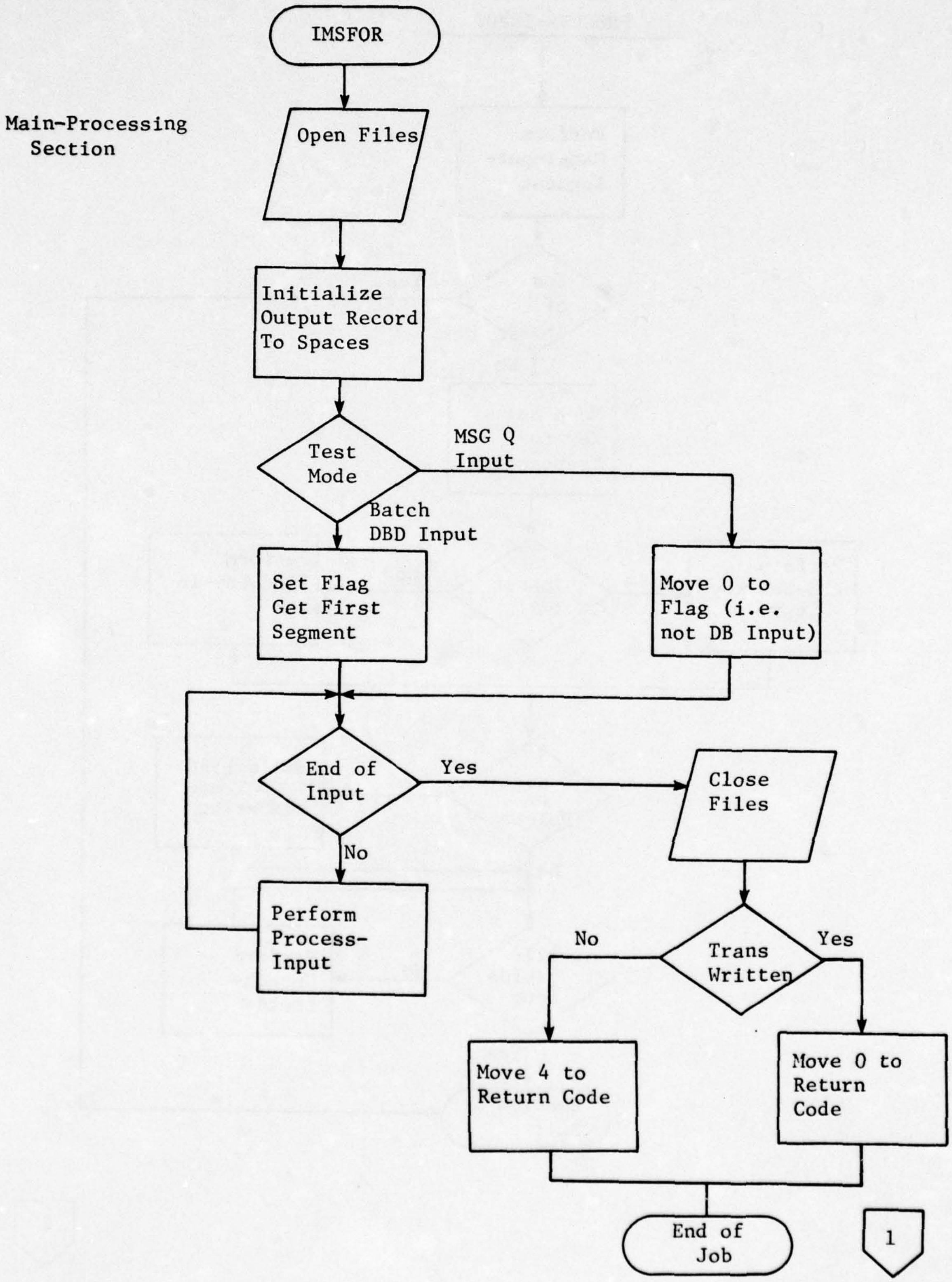
CHARACTER ENCODING:
HOL ☐ EXH ☐
BCD ☐ EBCDIC ☐ ASCII ☐
PACKED DECIMAL ☐
BINARY OR NON-CHARACTER INFORMATION ☐
OTHER ☐
PARITY: EVEN (7-TRACK BCD) ☐ ODD ☐

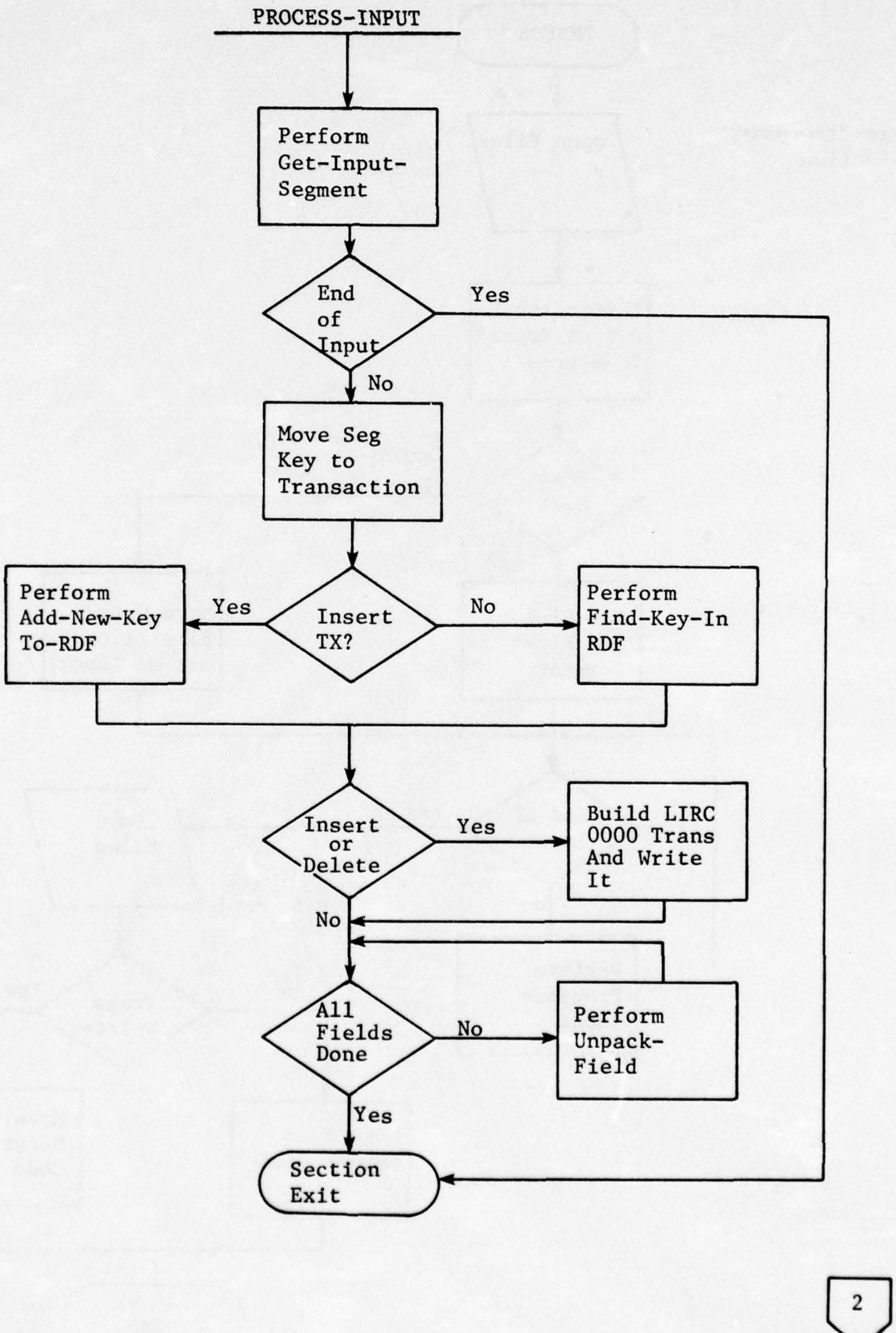
HEADER:
TRAILER:
COLLATING SEQ:

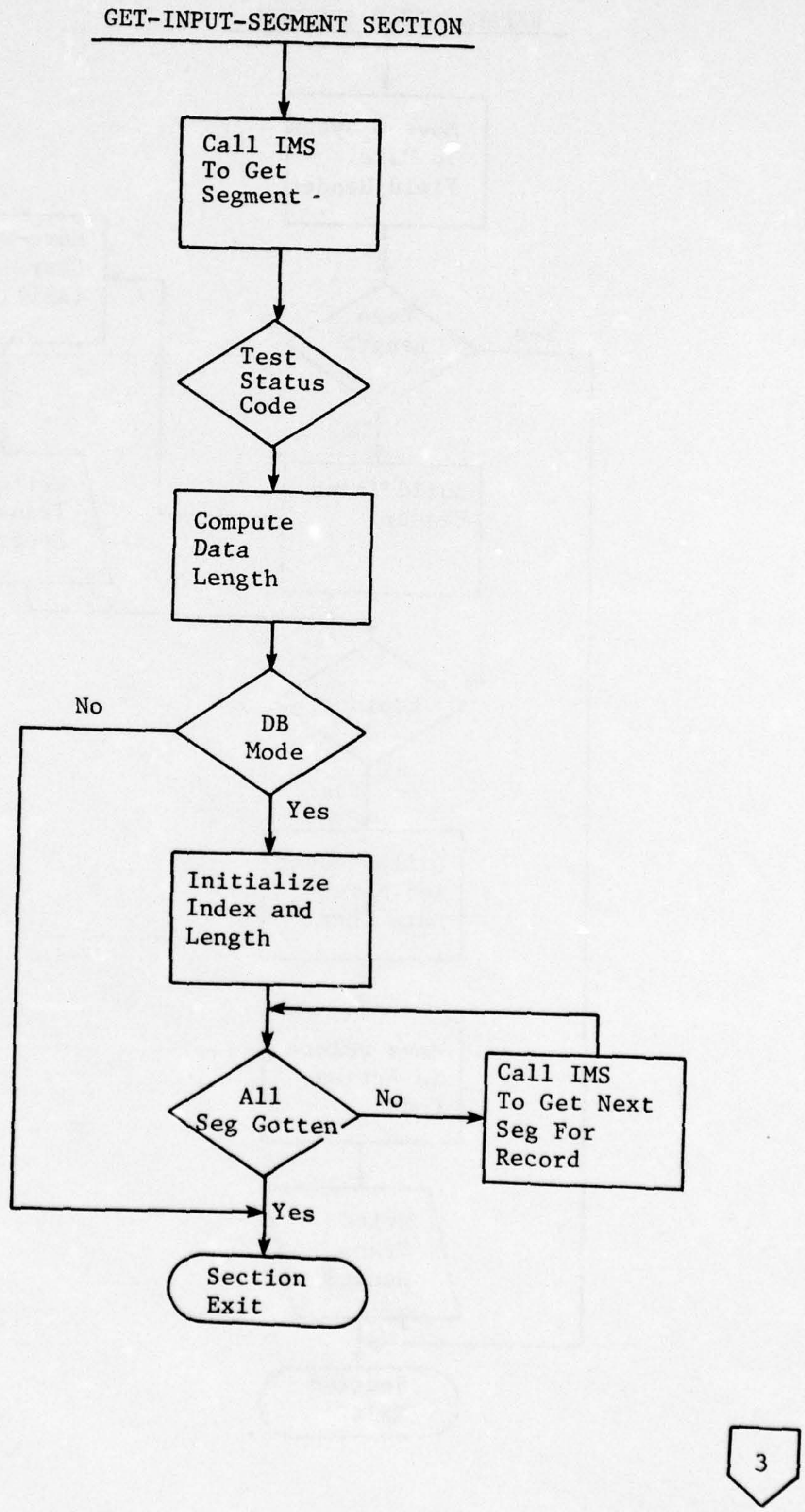
SEQUENCE OF RECORDS IN THE FILE		
	CONTROL FIELD TITLE	POSITIONS
MAJ		
2.		
3.		
4.		
5.		
MIN		

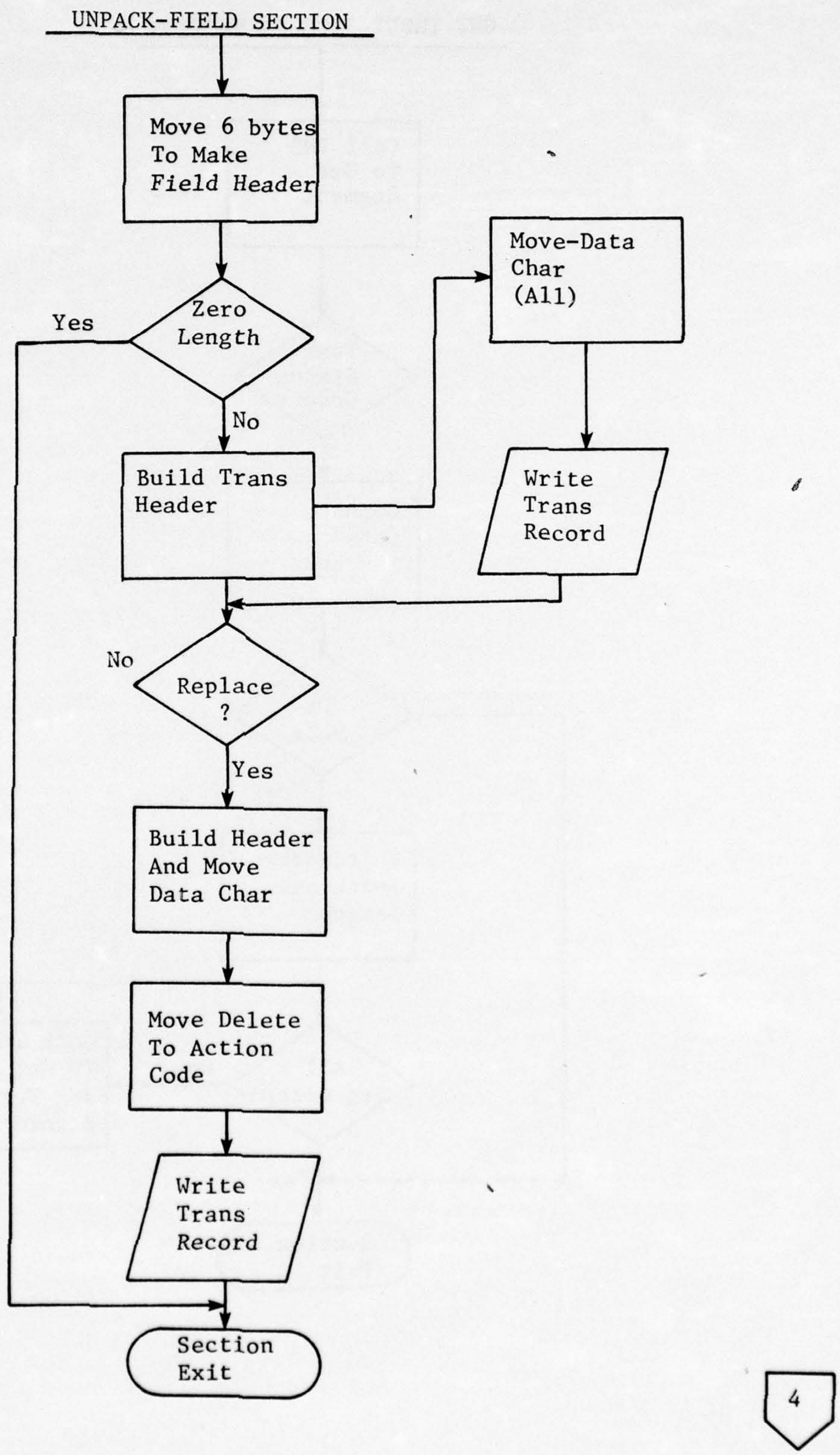
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

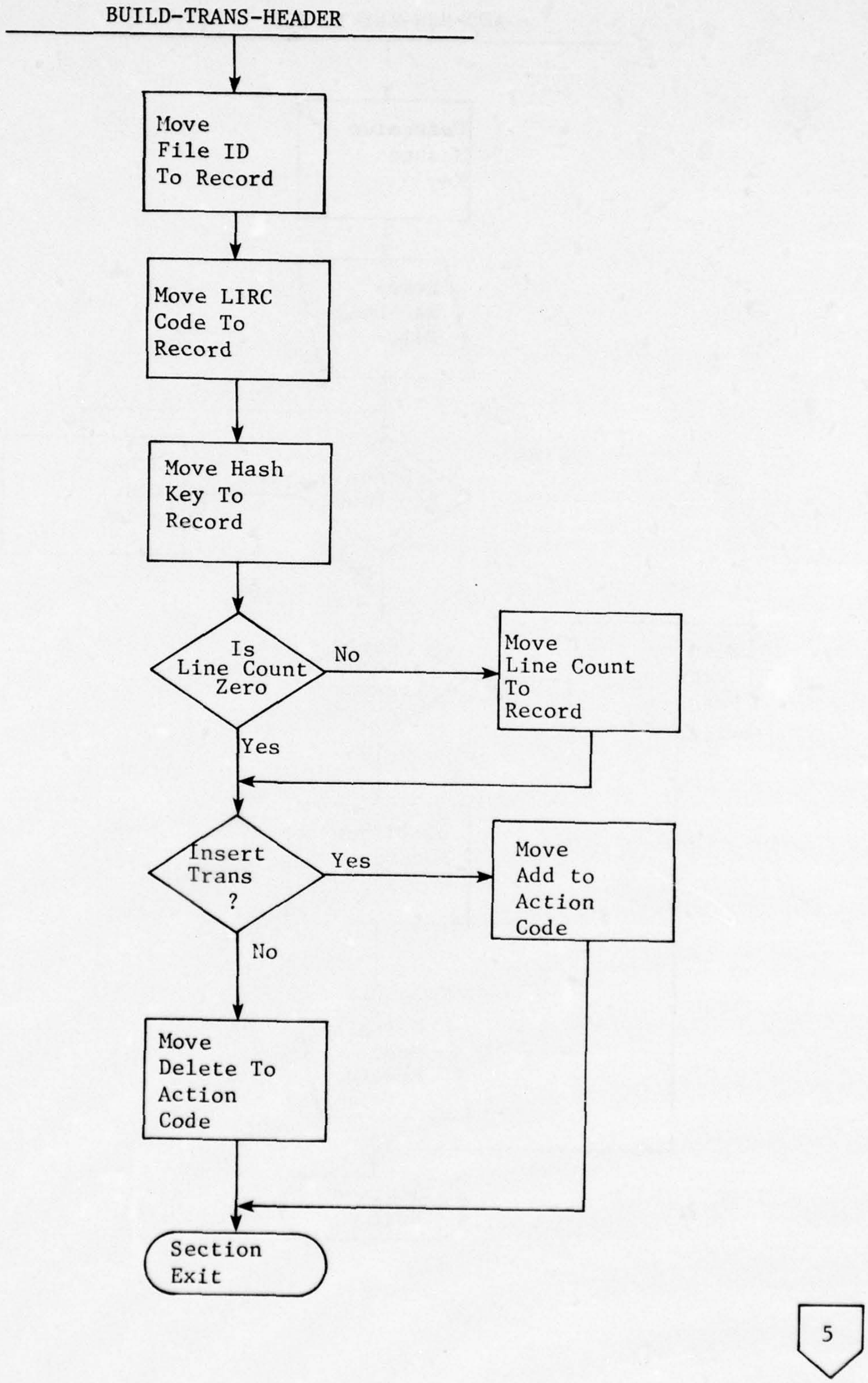
Figure A-18

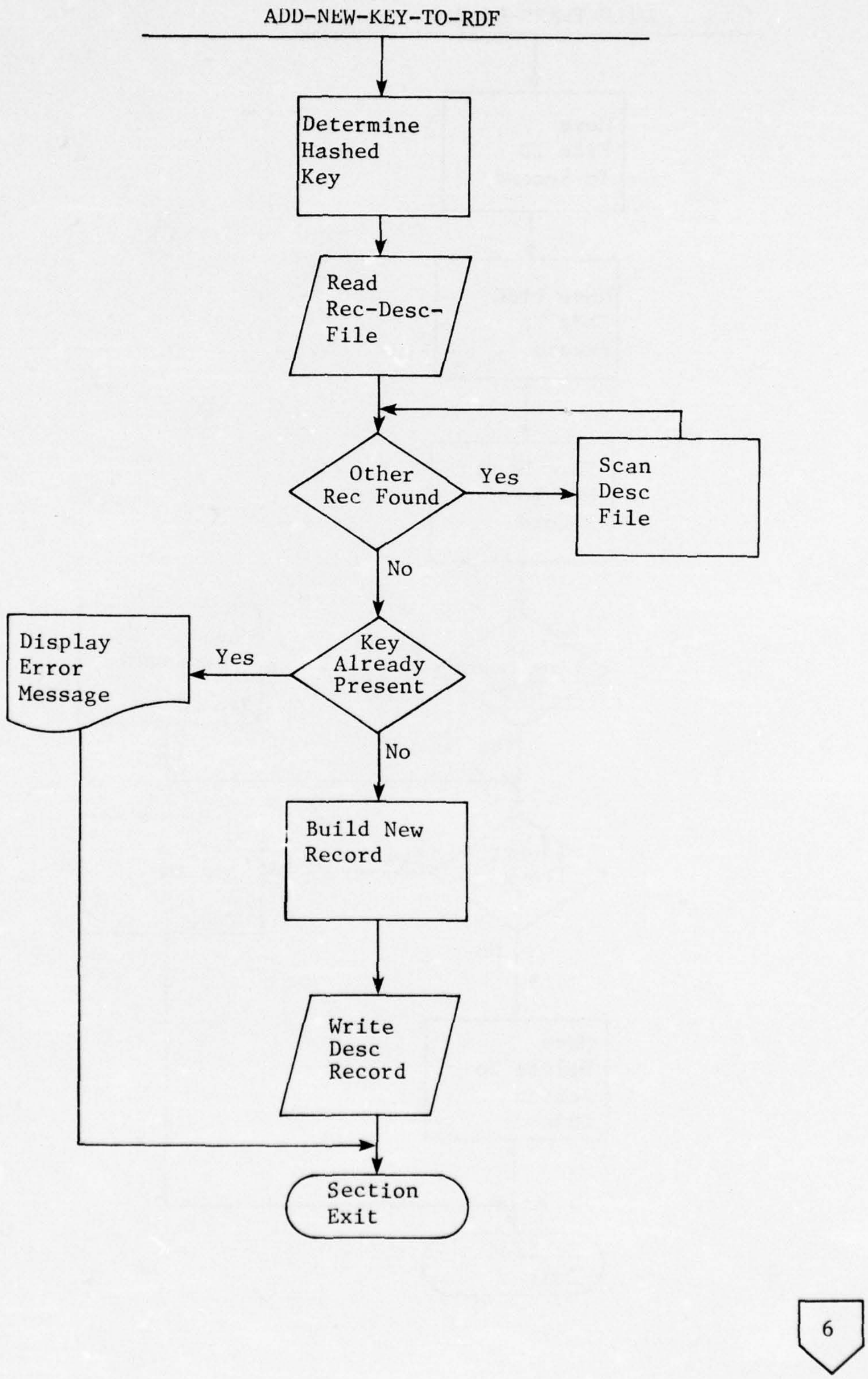

IMSFOR
Main-Processing
Section
Open Files
Initialize
Output Record
To Spaces
Test
Mode
MSG Q
Input
Batch
DBD Input
Set Flag
Get First
Segment
Move 0 to
Flag (i.e.
not DB Input)
End of
Input
Yes
No
Close
Files
Perform
Process-
Input
No
Trans
Written
Yes
Move 4 to
Return Code
Move 0 to
Return
Code
End of
Job
1

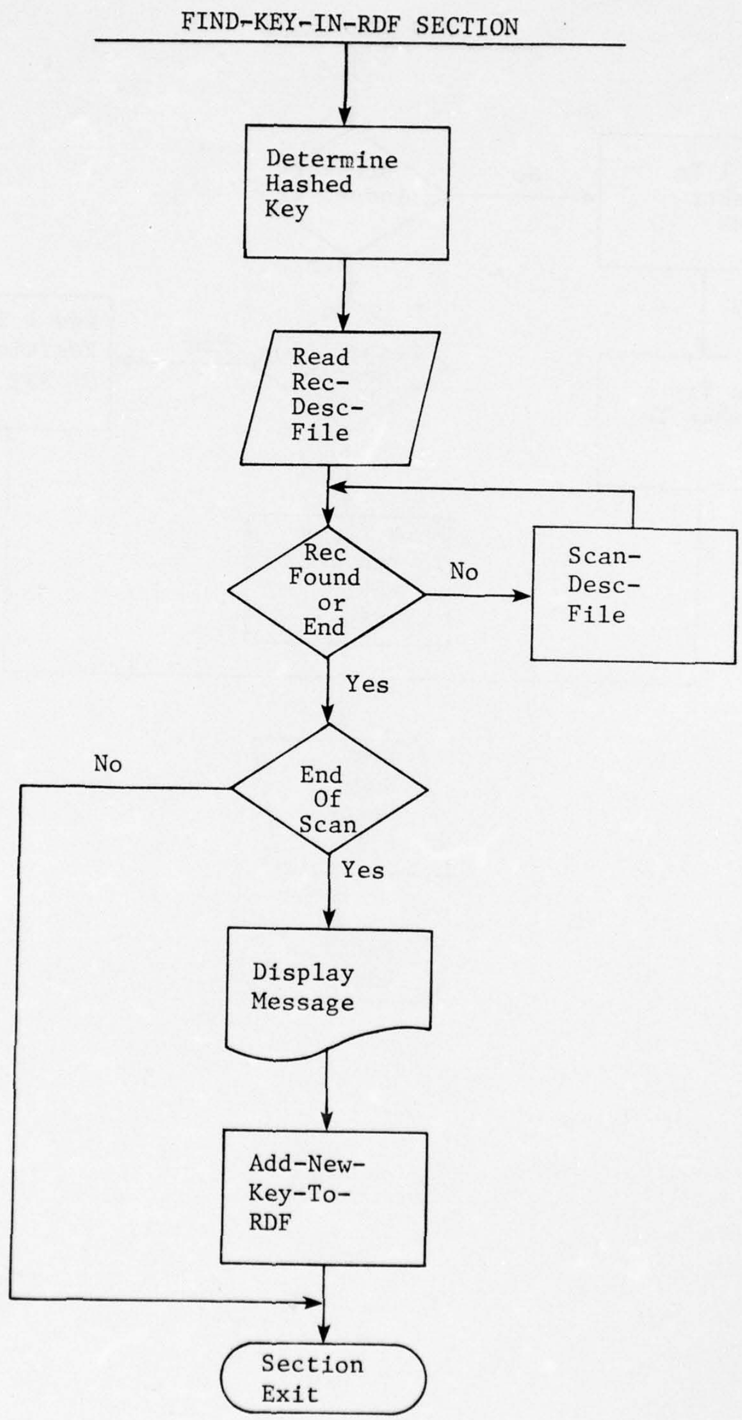

PROCESS-INPUT
Perform Get-Input-Segment
End of Input
Yes
No
Move Seg Key to Transaction
Perform Add-New-Key To-RDF
Yes
Insert TX?
No
Perform Find-Key-In RDF
Insert or Delete
Yes
Build LIRC 0000 Trans And Write It
No
All Fields Done
No
Perform Unpack-Field
Yes
Section Exit
2


GET-INPUT-SEGMENT SECTION
Call IMS To Get Segment -
Test Status Code
Compute Data Length
DB Mode
No
Yes
Initialize Index and Length
All Seg Gotten
No
Call IMS To Get Next Seg For Record
Yes
Section Exit
3

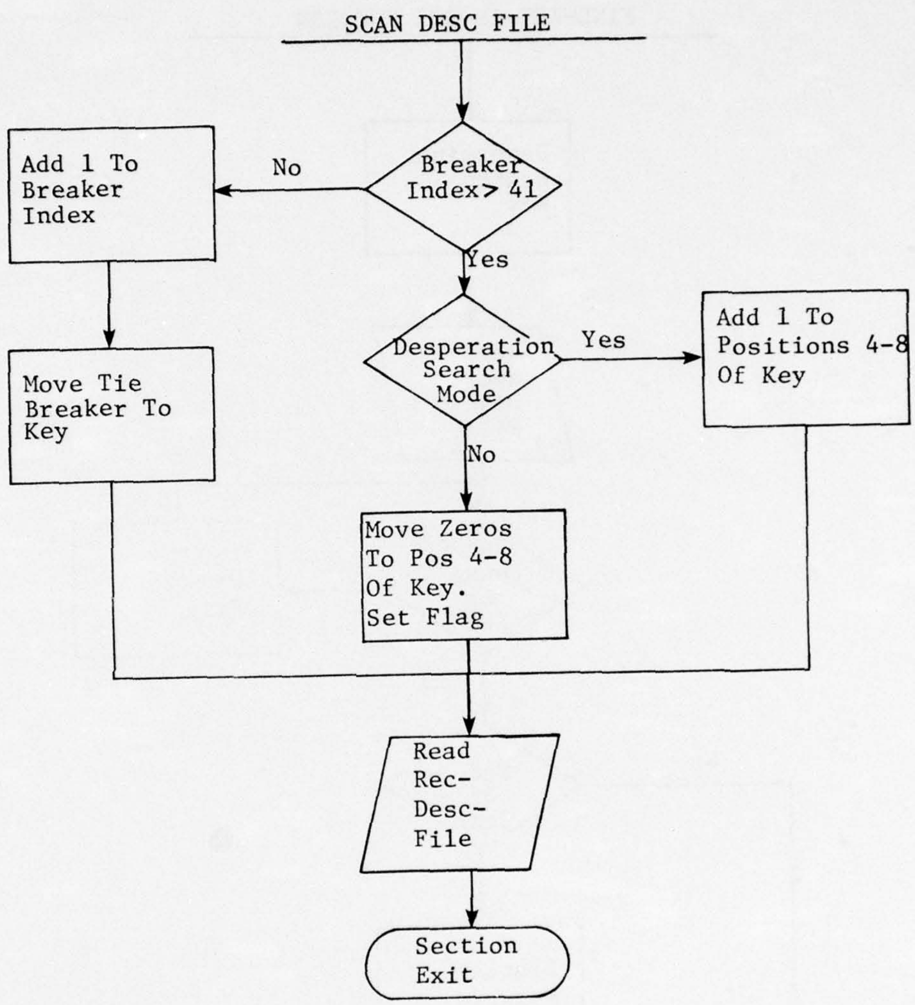

UNPACK-FIELD SECTION
Move 6 bytes To Make Field Header
Zero Length
Yes
No
Build Trans Header
Move-Data Char (All)
Write Trans Record
No
Replace ?
Yes
Build Header And Move Data Char
Move Delete To Action Code
Write Trans Record
Section Exit
4


BUILD-TRANS-HEADER
Move File ID To Record
Move LIRC Code To Record
Move Hash Key To Record
Is Line Count Zero
No
Move Line Count To Record
Yes
Insert Trans ?
Yes
Move Add to Action Code
No
Move Delete To Action Code
Section Exit
5


ADD-NEW-KEY-TO-RDF
Determine Hashed Key
Read Rec-Desc-File
Other Rec Found
Yes
Scan Desc File
No
Key Already Present
Yes
Display Error Message
No
Build New Record
Write Desc Record
Section Exit
6


FIND-KEY-IN-RDF SECTION
Determine Hashed Key
Read Rec-Desc-File
Rec Found or End
No
Scan-Desc-File
Yes
End Of Scan
No
Yes
Display Message
Add-New-Key-To-RDF
Section Exit

7


SCAN DESC FILE
Breaker Index > 41
No
Yes
Add 1 To Breaker Index
Move Tie Breaker To Key
Desperation Search Mode
Yes
No
Add 1 To Positions 4-8 Of Key
Move Zeros To Pos 4-8 Of Key. Set Flag
Read Rec-Desc-File
Section Exit

8

APPENDIX B

CINCLANTFLT NAVLIS APPLICATION CASE STUDY

I. INTRODUCTION

This report addresses the use of query sets in conjunction with inverted logistics data for the sharing (communication) of Navy logistics information. The base consists of ship scheduling and funding information. The report is presented to indicate the applicability of the NAVLIS concept to today's fleet problems. The application was developed with the assistance of the CINCLANTFLT Maintenance and Material Readiness Division, Code NFM3A.

II. BACKGROUND

Much of the early NAVLIS effort was concerned with an attempt to classify logistics information, synonym resolution, and dictionary schema. The LIRC code assignment problem had been studied.

After the project had been transferred to DTNSRDC, specifications for a standardized set of access routines (query modules) were written. These specifications were an adaptation of the already existing SHARP query modules. The NAVLIS query modules interfaced with network communications software for distributed system access. They also interfaced with the native computer system in order to access and retrieve the data from the base (files). Access is gained by means of an inverted file which was built and maintained in conjunction with the native system.

Although a NAVLIS users group was formed, the group was never instrumental in formulating requirements for a NAVLIS. It did ask questions. Because of the lack of direct user involvement in NAVLIS, the project manager with the assistance of his staff surveyed Fleet Maintenance at Norfolk and concluded that it would be a good logistics area in which to involve a user with NAVLIS.

A meeting between NAVLIS, 3M, and Fleet Maintenance personnel was held in Norfolk. 3M personnel maintained they had no need of NAVLIS involvement but the Fleet Maintenance user pointed out that the 3M system is lacking certain types of financial information which can be used in decision making.

NAVLIS personnel returned to DTNSRDC and built a small data base using the SHARP data base management system which could be queried from a terminal. (No computer instructions were written). A return visit was made to Norfolk and the user, during a single 1-1/2 hour interactive terminal session, defined his information requirement and obtained the computer's answer to his questions. Fleet Maintenance Division then, and greatly as a result of this experience agreed to cooperate in the NAVLIS research effort.

NAVLIS personnel again returned to Norfolk and performed a formal requirements analysis based on the user's assessment of his information access need. After completion of the requirements analysis, the analyst

returned to DTNSRDC and designed the data base architecture. The primary considerations in the mind of the analyst during the architectural design period were:

- The design would be open ended. No restrictions on the ability to add any type of logistics data to the base.
- No attempt would be made to physically account for data in the system. That function would be fully delegated to the SHARP data base management system.
- The design would be such that it would appear to the user that he was accessing distributed data base files. The architecture permitted this to be done in a single query, thus simulating NAVLIS.
- Information would be time referenced so that a tracing capability (pictures) of what was happening at a given time or during a specified period of time could be presented.
- A high degree of accuracy was required.
- The data base would not be classified.
- Accurate record keeping would be facilitated.

These design requirements were achieved through the following specific design features:

- The use of separate sets of descriptions within one description for different classes (files) of data so that additional sets can be added and/or incorporated at will without making any changes to the existing structure. Any type of data such as CASREP data can be added without disturbing the work done to date.
- The use "as-of-dates" to stamp <u>all</u> time-based information. The funding of a ship is an event. The dictionary in a sense is not.

- The use of redundancy to achieve accuracy. Individual accounting records contain balanced fields.
- Users can work <u>any time</u> in terms of ship names, UIC codes, or Hull designations. They may be mixed in the query in <u>any</u> combination desired.
- The information content of the base can be accessed on the basis of Navy organization qualifiers.
- The individual who is considered the "owner" is always available to query.
- There is no fuel information of any type in the base, no projected force levels, no information about ship activation and de-activation.

After the design was completed and the data base was built sufficiently to prove the design, a meeting took place in Norfolk. About two weeks later Fleet Maintenance personnel visited DTNSRDC, operated the system, and obtained the answers to questions.

III. OBJECTIVE

The NAVLIS objective in attacking ship scheduling and maintenance problems was two fold:

a. Research the information systems problem from the standpoint of on-line distributed data base architecture and access.

b. Apply the NAVLIS concept by doing something useful for the fleet using a scheme which involved the NAVLIS Pilot model.

It was recognized that resource management would be the most difficult field in which to apply NAVLIS because schedules and money are involved. It was also recognized where the pay off would be. Resource management necessitates the sharing (communication) of information.

The strategy used to meet the objective was to do what the fleet maintenance people wanted done, from a user's standpoint. Once the data base, which was solely for operational purposes on behalf of fleet maintenance, had been built, it would be moved to DPSCLANT where it could be queried without communication costs under SHARP or the NAVLIS Pilot Model installed locally at Norfolk. This would help DPSCLANT and using the NAVLIS tie-in between DPSCLANT and NMCSA, we would migrate the user from having the questions answered from the local base to having the same questions answered from the distributed files. Once this was achieved, the user would be in a position to validate the NAVLIS concept. After the model was in use, the Navy could decide the next step.

IV. APPLICATION

A. Information Content of the Base (See Figure B1 for SHARP definitions.)

The data base consists of approximately 4,000 records. The historical records in the data base include:

- overhaul start date
- end date
- type of overhaul
- yard
- home port
- commission date
- overhaul duration
- overhaul interval
- man days expended including new construction
- ship identification

The data dates back to approximately 1962.

The apportionment records in the base include the 73, 74, 75, and 76 fiscal years. The data include ship, start date, end-date man-day cost, material dollars, and unit cost.

The funding records for ship overhaul include the funding information for each ship for each month for the 12-month 75 fiscal year and July and

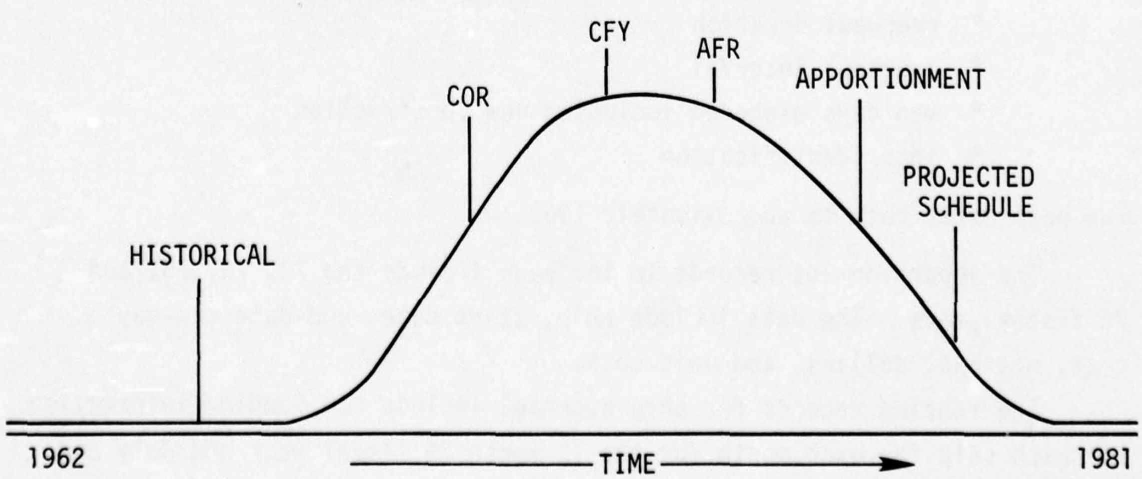
August of fiscal 76. Information includes:

1. ship identification
2. start date
3. end date
4. yard
5. complete set of funding information for active and reserve ships including:
   a. carry over requirements
   b. current fiscal year
   c. selected restricted availability
   d. advanced funding requirements

The data base also includes the overhaul standards for each class of ship, except those which are classified. The classified standards are not in the base.

The data base also includes a projected ship overhaul schedule out to 1981. This is very hypothetical and is not updated to reflect the current situation. It was included in the design as a tool for fleet maintenance operations use if and when "what if" questions are asked. This requires an algorithm. (See Figure B-1). This portion of the base consists of about 300 records.

To summarize, the data base consists of a set of duration and interval standards, a hypothetical (5-year forward) schedule, apportionment, advanced funding, current fiscal year, carry over funding activity, and ship overhaul history, all time stamped.



```
IO BASE2
0)RCDID; MAXSIZE=9;CHARS 1/5;NUMERIC OR VALUES = INTRO, LABEL,          0A
  DEFIN, INFOR, VALUS; CHARS 6/9; NUMERIC OR SPACES; INVERT             0B
1)NAVY;      MAXSIZE=15; INVERT                                         1A
2)PROC/DATE; SIZE=6; CHARS 1/2; VALUE=75; CHARS 3/4; RANGE=01/12;       2A
             CHARS 5/6; RANGE=01/31; INVERT                             2B
3)HULL;      MAXSIZE=8; INVERT                                          3A
4)SHIP;      MAXSIZE=18; INVERT                                         4A
5)UIC;       MAXSIZE=5; INTEGER; INVERT                                 5A
6)CMSNDATE; MAXSIZE=6; CHARS 1/2; RANGE=40/82; CHARS 3/4; RANGE=01/12;  6A
            CHARS 5/6; RANGE=01/31; INVERT                              6B
7)TYCOM;     MAXSIZE=1; INTEGER; INVERT                                 7A
8)ASOFDATE; MAXSIZE=6; CHARS 1/2; RANGE=50/82; CHARS 3/4; RANGE=01/12;  8A
           CHARS 5/6; RANGE=01/31; INVERT                               8B
9)COR; MAXSIZE=15; ALPHAB; INVERT                                       9A
10)DURATN; MAXSIZE=3; REAL; DEC 1; INVERT                               10A
11)INTRVL; MAXSIZE=3; REAL; DEC 1; INVERT                               11A
12)STARTDATE; MAXSIZE=6; CHARS 1/2; RANGE=50/82; CHARS 3/4;             12A
            RANGE=01/12; CHARS 5/6; RANGE=01/31; INVERT                 12B
13)ENDATE;   MAXSIZE=6; CHARS 1/2; RANGE=50/82; CHARS 3/4;              13A
            RANGE=01/12; CHARS 5/6; RANGE=01/31; INVERT                 13B
14)FCLTY; MAXSIZE=5; INVERT                                             14A
15)PYRYRFND; MAXSIZE=5; INTEGER; INVERT                                 15A
16)CHORD;    MAXSIZE=5; INTEGER; INVERT                                 16A
17)CNTYRPLN; MAXSIZE=5; INTEGER; INVERT                                 17A
18)CNTPLNTOT; MAXSIZE=5; INTEGER; INVERT                                18A
19)TOTFND;     MAXSIZE=5; INTEGER; INVERT                               19A
20)CNTYRFND;     MAXSIZE=5; INTEGER; INVERT                             20A
21)ESTCNTYRQT; MAXSIZE=5; INTEGER; INVERT                               21A
22)ESTOVHL;      MAXSIZE=5; INTEGER; INVERT                             22A
23)PERCNTOVHL; MAXSIZE=5; INTEGER; INVERT                               23A
24)BDGTLDAYS;   MAXSIZE=6; INTEGER; INVERT                              24A
25)BDGTLMNY;    MAXSIZE=5; INTEGER; INVERT                              25A
26)BDGTMMNY;    MAXSIZE=5; INTEGER; INVERT                              26A
27)BDGTUCOST;   MAXSIZE=5; INTEGER; INVERT                              27A
28)ACTLDAYS;    MAXSIZE=7; INTEGER; INVERT                              28A
29)ACTLMNY;     MAXSIZE=5; INTEGER; INVERT                              29A
30)ACTMMNY;     MAXSIZE=5; INTEGER; INVERT                              30A
31)ACTUCOST;    MAXSIZE=5; INTEGER; INVERT                              31A
32)HOMEPORT;    MAXSIZE=12; INVERT                                      32A
33)FNDACTVTY;   MAXSIZE=6; INVERT                                       33A
34)SCDACTVTY;   MAXSIZE=6; INVERT                                       34A
35)OVHLACTVTY; MAXSIZE=6; INVERT                                        35A
36)IMACTVTY;    MAXSIZE=6; INVERT                                       36A
37)TYPEOVHL;    MAXSIZE=6; INVERT                                       37A
38)PURCHORD;    MAXSIZE=6; INVERT                                       38A
39)LINKTO;      MAXSIZE=20; REPEAT; INVERT                              39A
40)NOTE;        MAXSIZE=1500                                            40A
41)ASSOC; MAXSIZE = 8; INVERT                                           41A
42)MARK;        SIZE=1; INVERT                                          42A
43)TWX;         MAXSIZE=1500                                            43A
44)LINKFM; MAXSIZE=20; REPEAT                                           44A
45)SHIPID; MAXSIZE=18; REPEAT; INVERT                                   45A
46)DATE;    MAXSIZE=6; CHARS 1/2; RANGE=50/82; CHARS 3/4; RANGE=01/12;  46A
            CHARS 5/6; RANGE=01/31; REPEAT; LINK; ROLE; INVERT          46B
47)MONEY;   MAXSIZE=10; INTEGER; REPEAT; LINK; ROLE; INVERT             47A
48)DICTSUBSET; MAXSIZE=15; INVERT                                       48A
49)ACRONYM; MAXSIZE=11; INVERT                                          49A
50)SECTION; MAXSIZE=2; ALPHAB; INVERT                                   50A
51)INFO; MAXSIZE=1500                                                   51A
52)NAVLIS; MAXSIZE=15; INVERT                                           52A
53)DICTSUBSET; MAXSIZE=15; INVERT                                       53A
54)SUBSETCTL; MAXSIZE=3; INTEGER; INVERT                                54A
55)HISTDATA; MAXSIZE=6; INVERT                                          55A
56)REPORT; MAXSIZE=3; INVERT                                            56A
61)SHIPS; MAXSIZE=30; REPEAT; INVERT                                    61A
```

Figure B-1

B. Queries and Reports

1. General

The scenarios which follow emphasize the ad hoc query capability of the pilot system. Pre-defined reports can be used to provide periodic reporting on a demand basis. The value added by an on-line capability to perform this pre-defined function to a community of users throughout the logistics environment is believed to be significant. The potential benefits of coupling a capability of this nature to a distributed set of comprehensive logistics data sets (files) should not be underestimated.

What follows in this section of the report is an in-depth presentation of the ability to access diversified sets of data using an ad hoc report definition and query capability. It is this ability to get questions answered, to find out what has happened and is happening in terms of a common English language which is easy to use and understand, that makes what follows so significant.

It is now possible to incorporate a wide range of diversified logistics data into an integrated structure and have the information available via a terminal. Approximately 50 definitions are contained in the base definition being demonstrated. Once the architect defines the expanded set of acronyms, data can be added or changed at will either from a remote terminal or by local batch operation.

It should be kept in mind that everything which follows in this section of the report was done with SHARP, but once developed, the NAVLIS Pilot Model will possess similar capabilities for truly distributed data bases.

The ability to access distributed sets of data from a single standardized query system is thoroughly demonstrated. The value added by such an approach is perhaps best demonstrated in the NAVLIS Scenario (see Attachment #6) where a projected ship schedule, budgeting and funding information, historic overhaul information, and overhaul duration and interval standards are presented in <u>one</u> NAVLIS report for the <u>entire</u> <u>Atlantic</u> <u>Fleet</u>.

2. LANTFLT Scenario I

LANTFLT Scenario I is an information scenario which presents budget (apportionment) data and actual funded costs for class DD, DDG, DLG, and DE ships. The information requirement originates in connection with a decision relating to implementation of fixed cost and 100% overhaul activity. The information is needed for each shipyard.

The scenario begins with a definition of information requirements in Report NF4. Query NF* is then formulated in conjunction with Report NF4 to retrieve the apportionment information for fiscal 74. Query NF* is repeated in the scenario for fiscal 75 and 76.

Query NF is then used in conjunction with Report NF4 to obtain the FY73 apportionment for the classes of interest. It is interesting to note the FY73 apportionment does not contain yard information.

LANTFLT Scenario I then obtains the actual funding for overhauls completed by first defining the information requirements in Report NF6 and then continuing with Query NF* to obtain the actual funding for completed overhauls in those classes of interest. Presentation of this information ends the scenario. See Attachment #1 for a representative report and terminal session.

3. Historical Scenario

This scenario displays information from the historical overhaul class of information available in the field of query.

A report NC1 is defined which presents the man days required to overhaul a ship. Since the first portion of the query set is concerned with new construction (NC), the start date and commission date are the event markers of interest. NC1 also presents sigma, maximum, minimum, average, and sum information.

Query NC1 presents information about SSBN new construction. Query QS1 uses Report NC1 and presents new construction information for FF class ships. A second question in Query QS1 asks for information about new construction on ARS class ships. The response

produces 29 hits for FF and zero hits for ARS, indicating that no new construction information for class ARS ships is available in the field of query.

As the ad hoc dialogue with the machine continues, Query QS2 asks three questions. The first two use Report NC1. The third uses another pre-defined report (NAV) contained in the system. The first question involves new construction for the LST class. The second question asks for all new construction which was performed in Philadelphia. The third question asks for all historic information about the ARS ships based on the information content of Report NAV, which sets forth the overhaul type and start and end dates for the overhaul, as well as the yard and man days. Notice the information content of this report contains no NC data, since there is none in the base for this class.

The ad hoc scenario continues with Query QS3. This query uses Report NAV but brings the overhauls down by start date in accordance with the sort criteria. It is interesting to see how the overhaul man days have increased over the years.

Query QS4 then brings the NAV ARS overhaul data out by sorting on hull and start date and subtotals on the actual labor days. This presents an interesting overhaul report for class ARS ships.

Query QS6 calls for all information contained in the data base concerning the BELKNAP. Notice no report format is specified. This results in a complete printout with full annotation of all information contained in the base. Query SET then terminates the session by printing the budget (apportionment) information contained in the base. Notice this information is identified as funding activity (fndactivity) by budget year. Since subtotals are taken on fndactivity and the fndactivity is also the primary sort key, the data are listed chronologically by budget years 73, 74, 75, and 76. See Attachment #2 for representative reports and terminal session.

4. <u>Ship Funding Scenario</u>

The ad hoc scenario starts by Query 003 asking the system to print the hull and ship of all ships which had advanced funding requirements (AFR) as of July 31, 1974 and a scheduled end date equal to or less than 7604 which is April of 76. We learn there were five such ships.

Since it is desired to trace the funding history (profile) of these five ships, Report 003 is defined to display the prior year funding, the current year plan, the current year funding, and the estimated current year requirement. The sub-totals of these fields are defined in the report. Query 004 then calls for all carry over requirement (COR), current fiscal year (CFY), selected restricted availability (SRA), or advanced funding requirement (AFR) type of information contained in the data base to be displayed in accordance with the information content of Report 003. Data are to be sorted down by hull and as-of date. The report demonstrates the year-end roll-over from current year funding to prior year funding. Sub-totals on this report are meaningless.

Query 005 of the Query set in this scenario then calls for the budget (apportionment) information on these ships for 74, 75, and 76 fiscal years. The ad hoc scenario then continues by defining a new information requirement in Report 005. Query 006 then, in conjunction with Report 005, displays the funding information on the same five ships on <u>one</u> report.

Query 007 then uses a pre-defined report PSB to display all information on these five ships in the base. Report PSB is sorted by hull and as-of date. Report PSB does not display the hull. It will be noticed that the first five digits of the RCDID are the ship UIC code. Query 008 simply asks the system to print the hull, ship, and UIC for the first record of each different 5-digit UIC record prefix. The printout of this information tells what hull is associated with each set of data in the PSB report. See Attachment #3 for representative reports and terminal session referred to in the preceding paragraphs.

5. Scenario Based on August 1975 Funding Report

The scenario begins by defining an information requirement in Report FND. This report demonstrates that a great deal of financial information can be presented on a standard ASR-33 type terminal. Report FLT is then defined as the second step in the scenario. This report sets forth scheduling information based on end date of the overhaul.

Query S03 then presents the funding situation by ship as of August 31, 1975. Report SCD (pre-defined) presents the basic hull, ship, year, start date, end date, and funding activity for that ship. The data are presented in the order of original entry by sorting on the record key suffix.

It is then recognized that Report FND has been improperly defined because the funding activity was not called for in the print statement. The error is corrected by simply re-defining Report FND. We then proceed with the scenario by using Query S03 to print the detailed funding information for each ship in the August 31, 1975 report in accordance with Report FND requirements. Although the data base contains all necessary cross footing, notice a 132-character printer (presently available at remote terminals) would really be required. This cross footing is displayed later.

Query S03 is then continued in the scenario by calling for all ships due to be completed in FY76 to be reported in accordance with the previously defined FLT Report. Data are to be sorted by end date (date ships leave the yard). The information is presented chronologically. See Attachment #4 for the representative reports and terminal session referred to in the preceding paragraphs.

6. Ship Funding Profile Scenario

This scenario commences with the definition of an information requirement. Information about how the estimates of a thorough overhaul vary during the overhaul, along with the estimates of the requirements for the present year and the actual funding for the

present year, is desired.

The analyst also needs to know how much money each yard has received by ship for fiscal 75. Sub-totals are taken. This information requirement is defined in the YRD Report as the second step in the ad hoc scenario.

Query *** then wants to know the hull and ship name of all ships which were reported as of January 31, 1975 as having started an actual overhaul (75 CFY) that month. Query *1* then calls for a funding profile of these ships starting July 31, 1974 through August 31, 1975. Report PRO presents the information in accordance with the sort requirements.

The scenario continues with Query *2*. This query asks the question and gets the answer to how the various yards were funded for that year.

The scenario continues with Query *3*. Pre-defined Report EST is used. This report prints the current year funding and the funding activity for ship, yard, start date, and end date. Query *3* is submitted again and the gate is widened to let all year end data associate and correlate in Report QRY. (To demonstrate the graceful degradation of the system and how the information is still not lost, another query is formulated, the 76 Budget data are included, and the 387 hits are printed on the computer room printer.) See Attachment #5 for representative reports and terminal session.

7. NAVLIS Scenario

The NAVLIS Scenario begins by an ad hoc information requirement consisting of 17 fields of information. For each ship the user would like to know the overhaul historical profile, the budgeting profile, the estimated overhaul cost for any actual funding which may have been recorded, and the relevant overhaul standards. The next hypothetical overhaul is also requested. All this information is to be appropriately time-sequenced on the printout.

The wealth of information which can be made available in a NAVLIS type system from a nucleus of approximately 50 elements is very large.

Much information in the base is not contained in this report. We could also ask questions about dates, home yards and overhaul yards, ships, etc. See Attachment #6 for representative reports and terminal session.

8. Ship Identification Scenario

This scenario demonstrates how the architecture of the system functions at the directory level.

Query DO asks 10 questions about ship identification. The power and flesibility of the system in its handling of information is demonstrated in the reports which are shown in Attachment #7. The reader should take special note: All the user is doing in Query DO is asking questions. The system automates the answer. Once the user has the ship names, hull, and UIC, he can ask any question using any of the three.

The *Ship Identification Scenario* is closed with a question about the Sunbird. The Sunbird is a member of the class ASR. The NAVLIS report tells us all about ASR ships. See Attachment #7 for representative reports and terminal session.

9. LANTFLT Scenario II

This scenario clearly demonstrates the ability of the user to obtain his information under less than optimum conditions. The scenario seeks information about LST and DE overhauls. It commences with Query WES and obtains a printout of the LST ships overhauled in Fiscal 75 and the first two months of Fiscal 76. Report WES is then defined to obtain information about the actual funding. The scenario continues with Query WES, which provides no hits because of the way in which not equal (NE) is used. Query WES is again formulated based only on the carry over requirement (COR) funding qualifier. Fifty-five hits are produced and printed in batch. Query WES then is input again, this time using all relevant funding qualifiers. Eighty-one hits are produced

and the user incorrectly calls for them on the terminal.

Query WES is aborted and again submitted and the output is taken in batch as it should be because of the 130-column report definition. The output presents a funding profile of the ship overhaul.

CINCLANTFLT

Scenario I

Terminal Session and Report Samples

Attachment #1

```
C 6700 INTERCOM V4.2
DATE  11/12/75
TIME  12.03.25.

PLEASE LOGIN
LOGIN,PUWEWEAVER,1180600823,SUP

COMMAND- ETL,500

COMMAND- COMRADE,SHARP



COMRADE       TIME:      12.04.23.
              DATE:      11/12/75

SHARP SUBSYSTEM ENTERED

???? REPDEF


 PLEASE INPUT REPORT DEFINITIONS.

ID BASE2
REPORT NF4
DBTYPE IS RCDID
MHEAD $APPORTIONMENT INFORMATION - ATLANTIC FLEET$
COL HEAD 1 = $YARD$
COL HEAD 2 = $HULL$
COL HEAD 3 = $SHIP$
COL HEAD 4 = $LABOR/MONEY/(1000)$
COL HEAD 5 = $MATL/MONEY/(1000)$
COL HEAD 6 = $UNIT/COST/(1000)$
PRINT FCLTY, HULL, SHIP, BDGTLMNY. BDGTMMNY, BDGTUCOST
PRINT SUBTOTAL BDGTLMNY, BDGTMMNY, BDGTUCOST ON FCLTY
PRINT TOTAL BDGTLMNY, BDGTMMNY, BDGTUCOST
PRINT MAX BDGTLMNY, BDGTMMNY, BDGTUCOST
PRINT MIN BDGTLMNY, BDGTMMNY, BDGTUCOST
PRINT SIGMA BDGTLMNY, BDGTMMNY, BDGTUCOST
$END


REPORT DEFINITION ERROR SUMMARY

ID - BASE2



REPORT - NF4
NO ERRORS FOUND - REPORT DEFINITION ACCEPTED
???? IQUERYX



YOU HAVE SELECTED THE INTERACTIVE QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-NO


SELECT ONE OF THE FOLLOWING DATA BASES
18 LOGISTICS DOCUMENT DATA BANK
31 YURSO DATA BANK
32 MKRS-BASE-2
?? 32


PLEASE INPUT QUERY COMMANDS
```

```
   LDV ERR 0003
   OVERLAY NOT FOUND
QUERY NF◆
IF FNDACTVTY IS 74BDGT?
AND PREFIX HULL IS DD?
AND PREFIX HULL IS NE TO DDG?
REPORT NF4
SORT ON FCLTY
HEAD 1 $DDACLASS - FISCAL 1974$
IF FNDACTVTY IS 74BDGT?
AND PREFIX HULL IS DDG?
REPORT NF4
SORT ON FCLTY
HEAD 1 $DDG CLASS - FISCAL 1974$
IF FNDACTVTY IS 74BDGT?
AND PREFIX HULL IS DLG?
REPORT NF4
SORT ON FCLTY
HEAD 1 $DLG CLASS - FISCAL 1974$
IF FNDACTVTY IS 74BDGT?
AND PREFIX HULL IS DE?
AND PREFIX HULL IS NE TO DEG?/
REPORT NF4
SORT ON FCLTY
HEAD 1 $DE CLASS - FISCAL 1974$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY NF◆, QUESTION NO.   1, HAS          5 HITS-T

QUERY NF◆, QUESTION NO.   2, HAS          4 HITS-T

QUERY NF◆, QUESTION NO.   3, HAS          4 HITS-T

QUERY NF◆, QUESTION NO.   4, HAS          2 HITS-T
```

APPORTIONMENT INFORMATION - ATLANTIC FLEET

DD CLASS - FISCAL 1974

REPORT NF4 NF*001 NOV 12 1975

YARD	HULL	SHIP	LABOR MONEY (1000)	MATL MONEY (1000)	UNIT COST (1000)	RCDID
PH	DD 820	RICH	2438	431	2869	521202678
		SUBTOTAL				
			2438	431	2869	
3	DD 835	CECIL	2590	428	3018	521352679
3	DD 839	POWER	2609	442	3051	521392680
3	DD 862	VOGELGESANG	2136	342	2478	521622681
3	DD 863	STEINAKER	2506	425	2931	521632682
		SUBTOTAL				
			9841	1637	11478	
		*****TOTAL****				
			12279	2068	14347	
		MAXIMUM*				
			2609	442	3051	
		MINIMUM*				
			2136	342	2478	
		****SIGMA*****				
			171	36	206	

APPORTIONMENT INFORMATION - ATLANTIC FLEET

DDG CLASS - FISCAL 1974

REPORT NF4   NF*002   NOV 12 1975

YARD	HULL	SHIP	LABOR MONEY (1000)	MATL MONEY (1000)	UNIT COST (1000)	RCDID
C	DDG 11	SELLERS	4277	626	4903	046772648
C	DDG 18	SEMMES	4391	653	5044	046842649
		SUBTOTAL				
			8668	1279	9947	
N	DDG 5	RICKETTS	3608	733	4341	046712643
		SUBTOTAL				
			3608	733	4341	
PH	DDG 6	BARNEY	3732	728	4460	046722637
		SUBTOTAL				
			3732	728	4460	
		*****TOTAL****				
			16008	2740	18748	
		MAXIMUM*				
			4391	733	5044	
		MINIMUM*				

LDV ERR 0003
OVERLAY NOT FOUND
DATA BASE - BASE2

	3608	626	4341
****SIGMA*****			
	337	47	294

APPORTIONMENT INFORMATION - ATLANTIC FLEET

DLG CLASS - FISCAL 1974

REPORT NF4 NF*003 NOV 12 1975

YARD	HULL	SHIP	LABOR MONEY (1000)	MATL MONEY (1000)	UNIT COST (1000)	RCDID
C	DLG 28	WAINWRIGHT	3533	592	4125	527032647
		SUBTOTAL				
			3533	592	4125	
N	DLG 27	DANIELS	2875	375	3250	527022642
		SUBTOTAL				
			2875	375	3250	
PH	DLG 6	FARRAGUT	5053	1225	6278	522312635
PH	DLG 17	YARNELL	3048	414	3462	526882636
		SUBTOTAL				
			8101	1639	9740	
		*****TOTAL****				
			14509	2606	17115	
		MAXIMUM*				
			5053	1225	6278	
		MINIMUM*				
			2875	375	3250	
		****SIGMA*****				
			858	341	1199	

APPORTIONMENT INFORMATION - ATLANTIC FLEET

DE CLASS - FISCAL 1974

REPORT NF4 NF*004 NOV 12 1975

YARD	HULL	SHIP	LABOR MONEY (1000)	MATL MONEY (1000)	UNIT COST (1000)	RCDID
C	DE 1049	KOELSCH	2601	542	3143	540442651
		SUBTOTAL				
			2601	542	3143	
PH	DE 1056	CANNOLE	2337	422	2759	540512638
		SUBTOTAL				
			2337	422	2759	
		*****TOTAL****				
			4938	964	5902	
		MAXIMUM*				
			2601	542	3143	
		MINIMUM*				
			2337	422	2759	
		****SIGMA*****				
			132	60	192	

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS

   LDV ERR 0003
   OVERLAY NOT FOUND
QUERY NF*
IF FNDACTVTY IS 75BDGT?
AND PREFIX HULL IS DD?
AND PREFIX HULL IS NE TO DDG?
REPORT NF4
SORT ON FCLTY
HEAD 1 $DD CLASS - FISCAL 1975$
IF FNDACTVTY IS 75BDGT?
AND PREFIX HULL IS DDG?
REPORT NF4
SORT ON FCLTY
HEAD 1 $DDG CLASS - FISCAL 1975$
IF FNDACTVTY IS 75BDGT?
AND PREFIX HULL IS DLG?
REPORT NF4
SORT ON FCLTY
HEAD 1 $DLG CLASS - FISCAL 1975$
IF FNDACTVTY IS 75BDGT?
AND PREFIX HULL IS DE?
AND PREFIX HULL IS NE TO DEG?
REPORT NF4
SORT ON FCLTY
HEAD 1 $DE CLASS - FISCAL 1975$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY NF*, QUESTION NO.   1, HAS        11 HITS-T

QUERY NF*, QUESTION NO.   2, HAS         5 HITS-T

QUERY NF*, QUESTION NO.   3, HAS         2 HITS-T

QUERY NF*, QUESTION NO.   4, HAS         4 HITS-T
```

APPORTIONMENT INFORMATION - ATLANTIC FLEET

DD CLASS - FISCAL 1975

REPORT NF4 NF◆001 NOV 12 1975

YARD	HULL	SHIP	LABOR MONEY (1000)	MATL MONEY (1000)	UNIT COST (1000)	RCDID
C	DD 937	DAVIS	6684	964	7648	521972548
C	DD 938	INGRAM	6626	976	7602	521982549
		◆◆◆SUBTOTAL◆◆◆				
			13310	1940	15250	
N	DD 941	DU PONT	6131	1023	7154	522002542
N	DD 943	BLANDY	5944	993	6937	522022543
		◆◆◆SUBTOTAL◆◆◆				
			12075	2016	14091	
PH	DD 714	RUSH	4782	686	5468	043142502
PH	DD 724	LAFFEY	4500	638	5138	043242500
PH	DD 820	RICH	4782	686	5468	521202501
PH	DD 940	MANLEY	6423	1030	7453	521992535
		◆◆◆SUBTOTAL◆◆◆				
			20487	3040	23527	
SJ	DD 890	MEREDITH	3816	735	4551	521902513
		◆◆◆SUBTOTAL◆◆◆				

LDV ERR 0003
OVERLAY NOT FOUND
DATA BASE - BASE2

			3816	735	4551	
3	DD 819	HOLDER	4450	715	5165	521192508
		◆◆◆SUBTOTAL◆◆◆				
			4450	715	5165	
8	DD 763	LAWE	3749	731	4480	038632510

APPORTIONMENT INFORMATION - ATLANTIC FLEET

DD CLASS - FISCAL 1975

REPORT NF4 NF*001 NOV 12 1975

YARD	HULL	SHIP	LABOR MONEY (1000)	MATL MONEY (1000)	UNIT COST (1000)	RCDID
		SUBTOTAL				
			3749	731	4480	
		*****TOTAL****				
			57887	9177	67064	
		MAXIMUM*				
			6684	1030	7648	
		MINIMUM*				
			3749	638	4480	
		****SIGMA*****				
			1067	152	1203	

APPORTIONMENT INFORMATION - ATLANTIC FLEET

DDG CLASS - FISCAL 1975

REPORT NF4 NF*002 NOV 12 1975

YARD	HULL	SHIP	LABOR MONEY (1000)	MATL MONEY (1000)	UNIT COST (1000)	RCDID
C	DDG 19	TATTNALL	7116	1161	8277	046852547
		SUBTOTAL				
			7116	1161	8277	
N	DDG 17	CONYNGHAM	6556	1189	7745	046832540
N	DDG 23	BYRD	6554	1187	7741	046902541
		SUBTOTAL				
			13110	2376	15486	
PH	DDG 2	ADAMS	6785	1211	7996	046682526
PH	DDG 3	KING	6812	1228	8040	046692527
		SUBTOTAL				
			13597	2439	16036	
		*****TOTAL****				
			33823	5976	39799	
		MAXIMUM*				
			7116	1228	8277	
		MINIMUM*				
			6554	1161	7741	
		****SIGMA*****				
			207	23	201	

APPORTIONMENT INFORMATION - ATLANTIC FLEET

DLG CLASS - FISCAL 1975

REPORT NF4 NF*003 NOV 12 1975

YARD	HULL	SHIP	LABOR MONEY (1000)	MATL MONEY (1000)	UNIT COST (1000)	RCDID
C	DLG 14	DEWEY	6070	948	7018	526852546
		SUBTOTAL				
			6070	948	7018	
N	DLG 34	BIDDLE	5597	895	6492	527092539
		SUBTOTAL				
			5597	895	6492	
		*****TOTAL****				
			11667	1843	13510	
		MAXIMUM*				
			6070	948	7018	
		MINIMUM*				
			5597	895	6492	
		****SIGMA*****				
			236	27	263	

APPORTIONMENT INFORMATION - ATLANTIC FLEET

DE CLASS - FISCAL 1975

REPORT NF4 NF*004 NOV 12 1975

YARD	HULL	SHIP	LABOR MONEY (1000)	MATL MONEY (1000)	UNIT COST (1000)	RCDID
C	DE 1047	VOGE	4954	761	5715	540422550
C	DE 1072	BLAKELY	4848	760	5608	540672551
		SUBTOTAL				
			9802	1521	11323	
PH	DE 1038	MCCLOY	2189	389	2578	540362528
PH	DE 1059	SIMS	4572	754	5325	540542529
		SUBTOTAL				
			6761	1143	7903	
		*****TOTAL****				
			16563	2664	19226	
		MAXIMUM*				
			4954	761	5715	
		MINIMUM*				
			2189	389	2578	
		****SIGMA*****				
			1135	160	1294	

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS

   LDV ERR 0003
   OVERLAY NOT FOUND
QUERY NF◆
IF FNDACTVTY IS 76BDGT?
AND PREFIX HULL IS DD?
AND PREFIX HULL IS NE TO DDG?
REPORT NF4
SORT ON FCLTY
HEAD 1 $DD CLASS - FISCAL 1976$
IF FNDACTVTY IS Ø6BDGT?
AND PREFIX HULL IS DDG?
REPORT NF4
SORT ON FCLTY
HEAD 1 $DDG CLASS - FISCAL 1976$
IF FNDACTVTY IS 76BDGT?
AND PREFIX HULL IS DLG?
REPORT NF4
SORT ON FCLTY
HEAD 1 $DLG CLASS - FISCAL 1976$
IF FNDACTVTY IS 76BDGT?
AND PREFIX HULL IS DE?
AND PREFIX HULL IS NE TO DEG?
REPORT NF4
SORT ON FCLTY

HEAD 1 $DE CLASS - FISCAL 1976$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY NF◆, QUESTION NO.   1, HAS          11 HITS-T

QUERY NF◆, QUESTION NO.   2, HAS           5 HITS-T

QUERY NF◆, QUESTION NO.   3, HAS           0 HITS
QUERY NF◆, QUESTION NO.   4, HAS           0 HITS
```

APPORTIONMENT INFORMATION - ATLANTIC FLEET

DD CLASS - FISCAL 1976

REPORT NF4 NF*001 NOV 12 1975

YARD	HULL	SHIP	LABOR MONEY (1000)	MATL MONEY (1000)	UNIT COST (1000)	PCDID
PH	DD 933	BARRY	9638	1714	11352	521935625
PH	DD 940	MANLEY	9628	1796	11424	521995621
		SUBTOTAL				
			19266	3510	22776	
SJ	DD 822	MCCARD	3911	918	4829	521225617
SJ	DD 866	CONE	3707	879	4586	521665615
		SUBTOTAL				
			7618	1797	9415	
3	DD 821	JOHNSTON	3911	852	4763	521215600
3	DD 842	FISKE	4489	1006	5495	521425601
3	DD 864	ELLISON	4281	906	5187	521645619
3	DD 871	DAMATO	4715	1079	5794	521715618
3	DD 880	DYESS	4613	1034	5647	521805602
3	DD 931	SHERMAN	8422	1825	10247	521915641
3	DD 943	BLANDY	8284	1807	10091	522025642
		SUBTOTAL				
			38715	8509	47224	
		*****TOTAL****				
			65599	13816	79415	
		MAXIMUM*				
			9638	1825	11424	
		MINIMUM*				
			3707	852	4586	
		****SIGMA*****				
			2341	406	2738	

APPORTIONMENT INFORMATION - ATLANTIC FLEET

DDG CLASS - FISCAL 1976

REPORT NF4 NF*002 NOV 12 1975

YARD	HULL	SHIP	LABOR MONEY (1000)	MATL MONEY (1000)	UNIT COST (1000)	RCDID
C	DDG 19	TATTNALL	10422	1720	12142	046855639
		SUBTOTAL				
			10422	1720	12142	
N	DDG 4	LAWRENCE	8973	1799	10772	046705630
N	DDG 23	BYRD	8869	1835	10704	046905629
		SUBTOTAL				
			17842	3634	21476	
PH	DDG 10	SAMPSON	9445	1867	11312	046765622
PH	DDG 38	LUCE	10254	2280	12534	522325623
		SUBTOTAL				
			19699	4147	23846	
		*****TOTAL****				
			47963	9501	57464	
		MAXIMUM*				
			10422	2280	12534	
		MINIMUM*				
			8869	1720	10704	
		****SIGMA*****				
			641	196	732	

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS

   LDV ERR 0003
   OVERLAY NOT FOUND


QUERY NF
IF SHIP IS SPARROW $DELETE
IF FNDACTVTY IS 73BDGT?
AND PREFIX HULL IS DD?
AND PREFIX HULL IS NE TO DDG?
REPORT NF4
HEAD 1 $DD CLASS - FISCAL 1973$
IF FNDACTVTY IS 73BDGT?
AND PREFIX HULL IS DDG?
REPORT NF4
HEAD 1 $DDG CLASS - FISCAL 1973$
IF FNDACTVTY IS 73BDGT?
AND PREFIX HULL IS DLG?
REPORT NF4
HEAD 1 $DLG CLASS - FISCAL 1973$
IF FNDACTVTY IS 73BDGT?
AND PREFIX HULL IS DE?
AND PREFIX HULL IS NE TO DEG?
REPORT NF4
HEAD 1 $DE CLASS - FISCAL 1973$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY NF , QUESTION NO.   1, HAS        10 HITS-T

QUERY NF , QUESTION NO.   2, HAS         0 HITS
QUERY NF , QUESTION NO.   3, HAS         3 HITS-T

QUERY NF , QUESTION NO.   4, HAS         1 HITS-T
```

APPORTIONMENT INFORMATION - ATLANTIC FLEET

DD CLASS - FISCAL 1973

REPORT NF4 NF 001 NOV 12 1975

YARD	HULL	SHIP	LABOR MONEY (1000)	MATL MONEY (1000)	UNIT COST (1000)	RCDID
	DD 822	MCCARD	2884	515	3399	521222613
	DD 841	NOA	1697	527	2224	521412588
	DD 850	KENNEDY	2606	522	3128	521502589
	DD 864	ELLISON	2549	522	3071	521642614
	DD 865	WARE	3047	545	3592	521652590
	DD 871	DAMATO	2828	566	3394	521712591
	DD 878	VESOLE	3047	545	3592	521782592
	DD 883	PERRY	2639	528	3167	521832593
	DD 942	BIGELOW	2965	538	3503	522012594
	DD 944	MULLINNIX	2718	560	3278	522032595

◆◆◆SUBTOTAL◆◆◆			
	26980	5368	32348
◆◆◆◆◆TOTAL◆◆◆◆			
	26980	5368	32348
◆◆◆MAXIMUM◆◆◆◆			
	3047	566	3592
◆◆◆MINIMUM◆◆◆◆			
	1697	515	2224
◆◆◆◆SIGMA◆◆◆◆◆			
	375	16	380

APPORTIONMENT INFORMATION - ATLANTIC FLEET

DLG CLASS - FISCAL 1973

REPORT NF4 NF 003 NOV 12 1975

YARD	HULL	SHIP	LABOR MONEY (1000)	MATL MONEY (1000)	UNIT COST (1000)	RCDID
	DLG 16	LEAHY	2888	591	3479	526872585
	DLG 28	WAINWRIGHT	3289	589	3878	527032586
	DLG 32	STANDLEY	3271	585	3856	527072587
		◆◆◆SUBTOTAL◆◆◆				
			9448	1765	11213	
		◆◆◆◆◆TOTAL◆◆◆◆				
			9448	1765	11213	
		◆◆◆MAXIMUM◆◆◆◆				
			3289	591	3878	
		◆◆◆MINIMUM◆◆◆◆				
			2888	585	3479	
		◆◆◆◆SIGMA◆◆◆◆◆				
			185	2	183	

APPORTIONMENT INFORMATION - ATLANTIC FLEET

DE CLASS - FISCAL 1973

REPORT NF4 NF 004 NOV 12 1975

YARD	HULL	SHIP	LABOR MONEY (1000)	MATL MONEY (1000)	UNIT COST (1000)	RCDID
	DE 1043	MCDONNELL	2155	416	2571	540392596

SUBTOTAL			
	2155	416	2571
*****TOTAL****			
	2155	416	2571
MAXIMUM*			
	2155	416	2571
MINIMUM*			
	2155	416	2571
****SIGMA*****			
	0	0	0

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- NO

???? REPDEF

 PLEASE INPUT REPORT DEFINITIONS.

ID BASE2
REPORT NF6
DBTYPE IS RCDID
MHEAD $FUNDING SUMMARY - ATLANTIC FLEET$
COL HEAD 1 = $YARD$
COL HEAD 2 = $HULL$
COL HEAD 3 = $END/DATE$
COL HEAD 4 = $START/DATE$
COL HEAD 5 = $EST/OVHL/(1000)$
COL HEAD 6 = $TOTAL/FNDNG/(1000)$
COL HEAD 7 = $PER/CENT/OVHL$
PRINT FCLTY, HULL, ENDATE, STARTDATE, ESTOVHL, TOTFND, PERCNTOVHL
PRINT SUBTOTAL ESTOVHL, TOTFND ON FCLTY
PRINT TOTAL ESTOVHL, TOTFND
PRINT MAX ESTOVHL TOTFND
PRINT MIN ESTOVHL, TOTFND
PRINT SIGMA ESTOVHL, TOTFND
$END

REPORT DEFINITION ERROR SUMMARY

ID - BASE2


REPORT - NF6
NO ERRORS FOUND - REPORT DEFINITION ACCEPTED
????
 EX ID=      CADC PFN=DATADEFFILE
 EX CY= 001  00001641 PRUS  $0004.10 /DAY
IQUERY


YOU HAVE SELECTED THE INTERACTIVE QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-NO


SELECT ONE OF THE FOLLOWING DATA BASES
18 LOGISTICS DOCUMENT DATA BANK
31 YURSO DATA BANK
32 MKRS-BASE-2
?? 32


PLEASE ENTER QUERY MODE OPTION.
ENTER A FOR ADVANCED, T FOR TUTORIAL-A

PLEASE INPUT QUERY COMMANDS
```

```
   LDV ERR 0003
   OVERLAY NOT FOUND
QUERY NF◆
IF ASOFDATE IS 750630?
AND ENDATE IS BE TO 74
AND PREFIX ENDATE IS BE TO 7$07/7506?
AND PREFIX HULL IS DD?
AND PREFIX HULL IS NE TO DDG?
REPORT NF6
SORT ON FCLTY
HEAD 1 $DD CLASS - FISCAL 75 COMPLETIONS$
IF ASOFDATE IS 750630?
AND ENDATE IS BE TO
AND PREFIX ENDATE IS BE TO 7407/7506?
AND PREFIX HULL IS DDG?
REPORT NF6
SORT ON FCLTY
HEAD 1 $DDG CLASS - FISCAL 75 COMPETIONS$
IF ASOFDATE IS 750630?
AND ENDATE IS BE TO 7407/7506?
AND PREFIX HULL IS DLG?
REPORT NF6
SORT ON FCLTY
HEAD 1 $DLG CLASS - FISCAL 75 COMPLETIONS$
IF ASOFDATE IS 750630?
AND ENDATE IS BE TO 7
AND PREFIX ENDATE IS BE TO 7407/7506?
AND PREFIX HULL IS DE?
AND PREFIX HULL IS NE TO DEG?
REPORT NF6
SORT ON FCLTY
HEAD 1 $DE CLASS - FISCAL 75 COMPETIONS$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY NF◆, QUESTION NO.   1, HAS           6 HITS-T

QUERY NF◆, QUESTION NO.   2, HAS           5 HITS-T

QUERY NF◆, QUESTION NO.   3, HAS           0 HITS
QUERY NF◆, QUESTION NO.   4, HAS           0 HITS
```

FUNDING SUMMARY - ATLANTIC FLEET

DD CLASS - FISCAL 75 COMPLETIONS

REPORT NF6 NF*001 NOV 12 1975

YARD	HULL	END DATE	START DATE	EST OVHL (1000)	TOTAL FNDNG (1000)	PER CENT OVHL	RCDID
PH	DD 820	JUN 75	JUN 74	6028	6028		521205341
			SUBTOTAL				
				6028	6028		
SJ	DD 890	JUN 75	JAN 75	4053	4053		521905348
			SUBTOTAL				
				4053	4053		
3	DD 839	OCT 74	MAR 74	4894	4894		521345343
3	DD 835	APR 75	AUG 74	3526	3526		521355344
3	DD 862	DEC 74	MAY 74	4026	4026		521625342
			SUBTOTAL				
				12446	12446		
8	DD 763	APR 75	AUG 74	4491	4491		038635347
			SUBTOTAL				
				4491	4491		
			*****TOTAL****				
				27018	27018		
			MAXIMUM*				
				6028	6028		
			MINIMUM*				
				3526	3526		
			****SIGMA*****				
				802	802		

FUNDING SUMMARY - ATLANTIC FLEET

DDG CLASS - FISCAL 75 COMPETIONS

REPORT NF6 NF♦002 NOV 12 1975

YARD	HULL	END DATE	START DATE	EST OVHL (1000)	TOTAL FNDNG (1000)	PER CENT OVHL	RCDID
C	DDG 18	DEC 74	JAN 74	5680	5680	100	046485227
C	DDG 11	MAY 75	JUN 74	6816	6816	100	046775226
			♦♦♦SUBTOTAL♦♦♦				
				12496	12496		
N	DDG 5	DEC 74	APR 74	6979	6979	100	046715224
			♦♦♦SUBTOTAL♦♦♦				
				6979	6979		
PH	DDG 6	FEB 75	APR 74	6710	6710	100	046725225
PH	DDG 37	JAN 75	NOV 73	9307	9307	100	522315222
			♦♦♦SUBTOTAL♦♦♦				
				16017	16017		
			♦♦♦♦♦TOTAL♦♦♦♦				
				35492	35492		
			♦♦♦MAXIMUM♦♦♦♦				
				9307	9307		
			♦♦♦MINIMUM♦♦♦♦				
				5680	5680		
			♦♦♦♦SIGMA♦♦♦♦♦				
				1195	1195		

CINCLANTFLT

Historical Scenario

Terminal Session and
Report Samples

Attachment #2

```
NSRDC 6700 INTERCOM V4.2
DATE  12/12/75
TIME  12.15.22.

PLEASE LOGIN
LOGIN,PUWEWEAVER,1180600823,SUP

COMMAND- ETL,500

COMMAND- COMRADE,SHARP


COMRADE        TIME:      12.16.12.
               DATE:      12/12/75

SHARP SUBSYSTEM ENTERED
???? IQUERY


YOU HAVE SELECTED THE INTERACTIVE QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-%A

???? REPDEF


 PLEASE INPUT REPORT DEFINITIONS.

ID BASE2
REPORT NC1
DBTYPE IS RCDID
MHEAD $NEW CONSTRUCTION MANDAY REPORTS
COL HEAD 1 = $HULL$
COL HEAD 2 = $START/DATE$
COL HEAD 3 = $CMSN/DATE$
COL HEAD 4 = $YARD$
COL HEAD 5 = $MAN/LABOR/(DAYS)$
PRINT HULL, STARTDATE, CMSNDATE, FCLTY, ACTLDAYS
PRINT SUM ACTLDAYS
PRINT SIGMA ACTLDAYS
PRINT AVERAGE ACTLDAYS
PRINT MAX ACTLDAYS
PRINT MIN ACTLDAYS
$END
```

```
EPORT DEFINITION ERROR SUMMARY

ID - BASE2


REPORT - NC1
NO ERRORS FOUND - REPORT DEFINITION ACCEPTED
???? IQUERY


YOU HAVE SELECTED THE INTERACTIVE QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-NO


SELECT ONE OF THE FOLLOWING DATA BASES
18 LOGISTICS DOCUMENT DATA BANK
31 YURSO DATA BANK
32 MKRS-BASE-2
?? 32

PLEASE INPUT QUERY COMMANDS
QUERY NC1
IF PREFIX HULL IS SSBN AND HISTDATA IS OVHL AND TYPEOVHL IS NC?
REPORT NC1
HEAD 1 $SHIP CLASS IS SSBN$
HEAD 2 $LISTED BY ASCENDING HULL$
SORT ON HULL
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY NC1, QUESTION NO.    1, HAS         29 HITS-T
```

NEW CONSTRUCTION MANDAY REPORT

SHIP CLASS IS SSBN

LISTED BY ASCENDING HULL

REPORT NC1 NC1001 DEC 12 1975

HULL	START DATE	CMSN DATE	YARD	MAN LABOR (DAYS)	RCDID
SSBN 602	AUG 13 58	MAR 01 61	PTSMH		051104410
SSBN 608	APR 01 59	MAY 15 61	GROT		051164412
SSBN 609		MAR 06 62	NEWS		051173331
SSBN 610	OCT 01 59	MAR 12 62	GROT		051184416
SSBN 611	DEC 14 59	MAY 21 62	NEWS		051194420
SSBN 616	SEP 01 60	APR 23 63	GROT	474250	051234424
SSBN 617	NOV 01 60	JUN 25 63	GROT	474250	051244431
SSBN 618	SEP 02 60	NOV 12 62	NEWS	469625	051254438
SSBN 620	FEB 06 61	MAY 18 64	PTSMH	779875	050624446
SSBN 622	APR 24 61	OCT 21 63	NEWS	522125	050744451
SSBN 623	MAY 01 61	OCT 08 63	GROT	527750	050754455
SSBN 625	JUL 10 61	FEB 28 64	NEWS	522125	050774468
SBN 627	OCT 23 61	JUN 26 64	NEWS	522125	057013336
SSBN 628	DEC 21 61	JUN 01 64	GROT	527750	057023482
SSBN 630	JAN 22 62	JUL 28 64	NEWS	660000	057043346
SSBN 631	MAR 01 62	AUG 04 64	GROT	660000	057053352
SSBN 633	MAY 01 62	JUN 29 64	GROT	660000	057074476
SSBN 635	MAR 14 62	OCT 27 64	NEWS	660000	057093491
SSBN 636	FEB 01 62	DEC 23 64	PTSMH	740832	057103496
SSBN 640	DEC 27 62	AUG 26 65	GROT	660000	057113502
SSBN 641	NOV 23 62	SEP 21 65	NEWS	660000	057123508
SSBN 643	MAR 28 63	JAN 21 66	GROT	660000	057143517
SSBN 644	JAN 07 63	DEC 22 65	NEWS	660000	057153523
SSBN 645	APR 03 63	APR 14 66	GROT	660000	057164484
SSBN 654	OCT 28 63	APR 29 66	NEWS	660000	057174491
SSBN 655	NOV 07 63	AUG 22 66	GROT	660000	057184497
SSBN 656	JAN 06 64	JUN 15 66	NEWS	660000	057194503
SSBN 657	MAR 16 64	NOV 28 66	NC	660000	057204482
SSBN 659	APR 13 64	APR 03 67	GROT	660000	057224518

*****TOTAL****

14800707

****AVERAGE***

616696

MAXIMUM*

779875

MINIMUM*

469625

NEW CONSTRUCTION MANDAY REPORT

SHIP CLASS IS SSBN

LISTED BY ASCENDING HULL

REPORT NC1 NC1001 DEC 12 1975

HULL	START DATE	CMSN DATE	YARD	MAN LABOR (DAYS)	RCDID
		****SIGMA*****			
				84940	

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES


PLEASE INPUT QUERY COMMANDS
QUERY QS1
IF PREFIX HULL IS FF AND HISTDATA IS OVHL AND TYPEOVHL IS NC?
REPORT NC1
SORT ON HULL
HEAD 1 $SHIP CLASS IS FF$
HEAD 2 $LISTED BY AZSCENDING HULL$
HEAD 3 $FIRST QUESTION IN THE QUERY SET$
IF PREFIX FULL IS ARS AND HISTDATA IS OVHL AND TYPEOVHL IS NC?
REPORT NC1

FATAL ERROR--INVALID SEARCH ACRONYM         --QUERY QS1, QUESTION 002
ABOVE QUESTION WAS NOT PROCESSED. PLEASE EXAMINE FOR SYNTAX ERROR
AND CONTENT, RE-ENTER IF POSSIBLE

IF PREFIX HULL IS ARS AND TYPEOVHL IS NC?
REPORT NC1
SORT ON HULL
HEAD 1 $SHIP CLASS IS ARS$
HEAD 2 $ORDER IS BE ASCENDING HULL$
HEAD 3 $SECOND QIUESTION IS N THE QUERY SET$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOREACH QUESTION, TYPE IN D, T, OR B.

QUERY QS1, QUESTION NO.   1, HAS         29 HITS-T

QUERY QS1, QUESTION NO.   2, HAS          0 HITS T
```

NEW CONSTRUCTION MANDAY REPORT

SHIP CLASS IS FF

LISTED BY ASCENDING HULL

FIRST QUESTION IN THE QUERY SET

REPORT NC1 QS1001 DEC 12 1975

HULL	START DATE	CMSN DATE	YARD	MAN LABOR (DAYS)	RCDID
FF 1038	JUN 26 61	OCT 21 63	08	71875	540363208
FF 1043	DEC 03 62	FEB 15 65	08	100000	540393780
FF 1044	DEC 03 62	AUG 05 65	08	100000	540403785
FF 1047	AUG 21 63	NOV 25 66	09	110000	540423213
FF 1049	SEP 03 63	MAY 23 67	09	110000	540443789
FF 1056	DEC 27 66	AUG 22 69	08	110000	540513794
FF 1059	DEC 27 66	JAN 03 70	08	110000	540543218
FF 1061	DEC 27 66	MAR 14 70	08	110000	540563798
FF 1068	DEC 27 66	JUN 13 70	08	110000	540633221
FF 1072	MAR 29 68	JUL 18 70	08	110000	540673224
FF 1075	MAR 29 68	SEP 19 70	08	110000	540703227
FF 1078	NOV 26 68	APR 24 71	08	110000	200493230
FF 1079	NOV 26 68	MAY 22 71	08	110000	200503233
FF 1080	JAN 23 69	AUG 14 71	08	110000	200513801
FF 1081	JAN 23 69	SEP 18 71	08	110000	200523804
FF 1082	JUL 25 69	OCT 30 71	08	110000	200533807
FF 1084	SEP 26 69	MAR 18 72	08	110000	200553810
FF 1085	SEP 26 69	JUL 22 72	08	110000	200563813
FF 1089	NOV 26 69	FEB 17 73	08	110000	200673816
FF 1090	NOV 26 69	MAR 31 73	08	110000	200683819
FF 1091	DEC 26 69	JUN 30 73	08	110000	200693822
FF 1092	DEC 26 69	JUL 28 73	08	110000	200703825
FF 1093	JAN 26 70	NOV 17 73	08	110000	200713828
FF 1094	APR 27 70	JAN 26 74	08	110000	200723831
FF 1096	MAY 11 70	JUL 27 74	08	110000	200743834
FF 1097	MAY 11 70	NOV 02 74	08	110000	200753836
FFG 4	JAN 06 64	APR 14 67	BATH	125000	046953838
FFG 5	JAN 06 64	JUL 27 67	BATH	125000	046983236
FFG 6	JAN 06 64	NOV 03 67	BATH	125000	046993240

*****TOTAL****

3176875

****AVERAGE***

109547

MAXIMUM*

125000

NEW CONSTRUCTION MANDAY REPORT

SHIP CLASS IS FF

LISTED BY ASCENDING HULL

FIRST QUESTION IN THE QUERY SET

REPORT NC1 QS1001 DEC 12 1975

LDV ERR 0003
OVERLAY NOT FOUND
DATA BASE - BASE2

HULL	START DATE	CMSN DATE	YARD	MAN LABOR (DAYS)	RCDID
		MINIMUM*			
				71875	
		****SIGMA*****			
				8949	

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?-

PLEASE ANSWER YES OR NO- YES

PLEASE INPUT QUERY COMMANDS
QYEUERY QS2
IF PREFIX HULL IS LST AND HISTDATA IS OVHL AND TYPEOVHL IS NC?
REPORT NC1
SORT ON HULL
HEAD 1 $SHIP CLASS IS LST$
HEAD 2 $ORDER IS BY ASCENDING HULL$
HEAD 3 %FIRST QUESTION IN THE QUERY SET$
HEAD 3 $FIRST QUESTION IN THE QUERY SET$
IF FCLTY IS $$DELETE
IF PREFIX FCLTY IS PH AND TYPEOVHL IS NC?
REPORT NC1
SORT ON STARTDATE
HEAD 1 $SHIPYARD IS PHILA$
HEAD 2 $WORKLOAD ASSIGNMENT ALALYSIS$
HEAD 3 $SECOND QUESTION IN THE QUERY SET$
IF PREFIX HULL IS ARS AND HISTDATA IS OVHL?
REPORT NAV
SORT ON FCLTY AND HULL
HEAD 1 $SHIP CLASS IS ARS$
HEAD 2 $ORDER IS BY FCLTY AND HULL$
HEAD 3 $THRIRD QUESTION IN THE QUERY SET$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY QS2, QUESTION NO.   1, HAS          9 HITS-T

QUERY QS2, QUESTION NO.   2, HAS          6 HITS-T

QUERY QS2, QUESTION NO.   3, HAS         16 HITS-T
```

NEW CONSTRUCTION MANDAY REPORT

SHIP CLASS IS LST

ORDER IS BY ASCENDING HULL

FIRST QUESTION IN THE QUERY SET

REPORT NC1 QS2001 DEC 12 1975

HULL	START DATE	CMSN DATE	YARD	MAN LABOR (DAYS)	RCDID
LST 1179	JUL 05 66	SEP 02 69	PHILA	402486	581793965
LST 1180	SEP 06 66	JAN 24 70	PHILA	344757	200193315
LST 1181	OCT 03 66	JUN 20 70	PHILA	337020	200203318
LST 1188	FEB 03 69	JAN 23 71	SD	225000	200273320
LST 1192	NOV 17 69	SEP 01 71	11	225000	200313323
LST 1193	JAN 30 70	OCT 16 71	11	225000	200323969
LST 1194	MAR 06 70	DEC 18 71	11	225000	200333972
LST 1196	AUG 03 70	APR 08 72	11	225000	202223975
LST 1197	SEP 08 70	MAY 27 72	11	225000	202233978

*****TOTAL****

2434263

****AVERAGE***

270474

MAXIMUM*

402486

MINIMUM*

225000

****SIGMA*****

66484

NEW CONSTRUCTION MANDAY REPORT

SHIPYARD IS PHILA

WORKLOAD ASSIGNMENT ALALYSIS

SECOND QUESTION IN THE QUERY SET

REPORT NC1 QS2002 DEC 12 1975

HULL	START DATE	CMSN DATE	YARD	MAN LABOR (DAYS)	RCDID
DDG 44	NOV 01 57	OCT 19 61	PHILA		526843909
LPH 7	NOV 01 60	SEP 30 63	PHILA	528025	073523288
LPH 9	APR 02 62	JAN 16 65	PHILA	499375	071783294
LST 1179	JUL 05 66	SEP 02 69	PHILA	402486	581793965
LST 1180	SEP 06 66	JAN 24 70	PHILA	344757	200193315
LST 1181	OCT 03 66	JUN 20 70	PHILA	337020	200203318

*****TOTAL****

2111663

****AVERAGE***

422333

MAXIMUM*

528025

MINIMUM*

337020

****SIGMA*****

78481

DATA BASE - BASE2

SHIPYARD WORK ASSIGNMENT REPORT

SHIP CLASS IS ARS

ORDER IS BY FCLTY AND HULL

THIRD QUESTION IN THE QUERY SET

REPORT NAV 0S2003 DEC 12 1975

FCLTY	HULL	START DATE	END DATE	TYPE OVHL	MAN DAYS
JACK	ARS 6	JUN 08 70	JAN 17 71	RO	7215
JACK	ARS 6	JUN 15 75	NOV 17 75	RO	11315
PHILA	ARS 40	FEB 10 71	JUL 02 71	RO	21022
03	ARS 43	AUG 14 62	OCT 26 62	RO	2564
03	ARS 6	MAR 07 61	JUN 16 61	RO	
04	ARS 40	SEP 10 63	JAN 13 64	RO	4650
05	ARS 40	FEB 13 67	JUN 20 67	RO	7390
05	ARS 40	SEP 24 74	MAR 18 75	RO	15455
05	ARS 41	SEP 13 68	FEB 17 69	RO	7340
05	ARS 41	JUN 16 72	JAN 08 73	RO	9255
05	ARS 43	JAN 03 66	APR 12 66	RO	5590
05	ARS 43	JUN 27 69	OCT 30 69	RO	10517
05	ARS 43	APR 04 73	OCT 24 73	RO	12250
05	ARS 6	MAY 20 63	AUG 20 63	RO	2480
06	ARS 41	APR 16 65	AUG 13 65	RO	8510
10	ARS 6	SEP 20 66	APR 28 67	RO	9600

*****TOTAL****

135153

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

QUERY RUN ABORTED IN ROUTINE LT
 REPDEF3 - OPEN OLD ON A NON-EXISTING FILE
  SOURCE LINE NUMBER 00556
  FATAL COBOL ERROR - RUN TERMINATED

WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS
QUERY QS3

IF PREFIX HULL IS ARS AND HISTDATA IS OVHL?
REPORT NAV
SORT ON STARTDATE
HEAD 1 $SHIP CLASS IS ARS$
HEAD 2 $OVERHAUL HISTORY ORDERED BY START DATE$
HEAD 3 $ADDITIONAL INFORMATION IS REQUIRED AT QUERY TIME$
PRINT AVERAGE ACTLDAYS
PRINT MAX ACTLDAYS
PRINT MIN ACTLDAYS
PRINT SIGMA ACTLDAYS
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY QS3, QUESTION NO.    1, HAS         16 HITS-T
```

SHIPYARD WORK ASSIGNMENT REPORT

SHIP CLASS IS ARS

OVERHAUL HISTORY ORDERED BY START DATE

ADDITIONAL INFORMATION IS REQUIRED AT QUERY TIME

REPORT NAV QS3001 DEC 12 1975

FCLTY	HULL	START DATE	END DATE	TYPE OVHL	MAN DAYS
03	ARS 6	MAR 07 61	JUN 16 61	RO	
03	ARS 43	AUG 14 62	OCT 26 62	RO	2564
05	ARS 6	MAY 20 63	AUG 20 63	RO	2480
04	ARS 40	SEP 10 63	JAN 13 64	RO	4650
06	ARS 41	APR 16 65	AUG 13 65	RO	8510
05	ARS 43	JAN 03 66	APR 12 66	RO	5590
10	ARS 6	SEP 20 66	APR 28 67	RO	9600
05	ARS 40	FEB 13 67	JUN 20 67	RO	7390
05	ARS 41	SEP 13 68	FEB 17 69	RO	7340
05	ARS 43	JUN 27 69	OCT 30 69	RO	10517
JACK	ARS 6	JUN 08 70	JAN 17 71	RO	7215
PHILA	ARS 40	FEB 10 71	JUL 02 71	RO	21022
05	ARS 41	JUN 16 72	JAN 08 73	RO	9255
05	ARS 43	APR 04 73	OCT 24 73	RO	12250
5	ARS 40	SEP 24 74	MAR 18 75	RO	15455
JACK	ARS 6	JUN 15 75	NOV 17 75	RO	11315

*****TOTAL****

135153

****AVERAGE***

9010

MAXIMUM*

21022

MINIMUM*

2480

****SIGMA*****

4677

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS
QUERY QS4
IF PREFIX HULL IS ARS AND HISTDATA IS OVHL?
REPORT NAV
SORT ON HULL AND STARTDATE
HEAD 1 $SHIP CLASS IS ARS$
HEAD 2 $HISTORY OF EACH SHIP$
HEAD 3 $THE INFORMATION PROCESSING POWER OF NAVLIS$
PRINT SUBTOTAL ACTLDAYS ON HULL
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY QS4, QUESTION NO.    1, HAS          16 HITS-T
```

SHIPYARD WORK ASSIGNMENT REPORT

SHIP CLASS IS ARS

HISTORY OF EACH SHIP

THE INFORMATION PROCESSING POWER OF NAVLIS

REPORT NAV QS4001 DEC 12 1975

FCLTY	HULL	START DATE	END DATE	TYPE OVHL	MAN DAYS
04	ARS 40	SEP 10 63	JAN 13 64	RO	4650
05	ARS 40	FEB 13 67	JUN 20 67	RO	7390
PHILA	ARS 40	FEB 10 71	JUL 02 71	RO	21022
05	ARS 40	SEP 24 74	MAR 18 75	RO	15455
		SUBTOTAL			
					48517
06	ARS 41	APR 16 65	AUG 13 65	RO	8510
05	ARS 41	SEP 13 68	FEB 17 69	RO	7340
5	ARS 41	JUN 16 72	JAN 08 73	RO	9255
		SUBTOTAL			
					25105
03	ARS 43	AUG 14 62	OCT 26 62	RO	2564
05	ARS 43	JAN 03 66	APR 12 66	RO	5590
05	ARS 43	JUN 27 69	OCT 30 69	RO	10517
05	ARS 43	APR 04 73	OCT 24 73	RO	12250
		SUBTOTAL			
					30921
03	ARS 6	MAR 07 61	JUN 16 61	RO	
05	ARS 6	MAY 20 63	AUG 20 63	RO	2480
10	ARS 6	SEP 20 66	APR 28 67	RO	9600
JACK	ARS 6	JUN 08 70	JAN 17 71	RO	7215
JACK	ARS 6	JUN 15 75	NOV 17 75	RO	11315
		SUBTOTAL			
					30610

SHIPYARD WORK ASSIGNMENT REPORT

SHIP CLASS IS ARS

HISTORY OF EACH SHIP

THE INFORMATION PROCESSING POWER OF NAVLIS

REPORT NAV QS4001 DEC 12 1975

FCLTY	HULL	START DATE	END DATE	TYPE OVHL	MAN DAYS
		*****TOTAL****			
					135153

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?-
 DATA BASE - BASE2          YES

PLEASE INPUT QUERY COMMANDS
QUERY QS6
IF SHIP IS BELKNAP?
SORT ON ASOFDATE
HEAD 1 $THE POWER OF NAVLIS TO AUDIT$
HEAD 2 $FULL DISCLOSURE ON THE BELKNAP$
HEAD 3 $DEMONSTRATION OF THE INTEGRATION OF THE ARCHITECTURE$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY QS6, QUESTION NO.   1, HAS          9 HITS-T
```

THE POWER OF NAVLIS TO AUDIT

FULL DISCLOSURE ON THE BELKNAP

DEMONSTRATION OF THE INTEGRATION F THE ARCHITECTURE

REPORT DFT 0S6001 DEC 12 1975

ACTLDAYS: 288125
ASOFDATE: 611002
CDR: HENCK
CMSNDATE: 641107
DURATN: 3.0
ENDATE: 641104
FCLTY: BATH
HISTDATA: OVHL
HULL: CG 26
INTRVL: 33.0
NAVY: NAVSEA
PROC/DATE: 751114
RCDID: 527013928
SHIP: BELKNAP
STARTDATE: 611002
TYCOM: 2
TYPEOVHL: NC
UIC: 52701

ACTLDAYS: 7960
ASOFDATE: 641109
CDR: HENCK
CMSNDATE: 641107
DURATN: 4.0
ENDATE: 650205
FCLTY: BSN
HISTDATA: OVHL
HULL: CG 26
INTRVL: 33.0
NAVY: NAVSEA
PROC/DATE: 751114
RCDID: 527013929
SHIP: BELKNAP
STARTDATE: 641109
TYCOM: 2
TYPEOVHL: FO
UIC: 52701

THE POWER OF NAVLIS TO AUDIT

FULL DISCLOSURE ON THE BELKNAP

DEMONSTRATION OF THE INTEGRATION OF THE ARCHITECTURE

REPORT DFT QS6001 DEC 12 1975

ACTLDAYS: 18398
ASOFDATE: 650730
CDR: HENCK
CMSNDATE: 641107
DURATN: 4.0
ENDATE: 651126
FCLTY: NORVA
HISTDATA: OVHL
HULL: CG 26
INTRVL: 33.0
NAVY: NAVSEA
PROC/DATE: 751114
RCDID: 527013930
SHIP: BELKNAP
STARTDATE: 650730
YCOM: 2
YPEOVHL: PS
UIC: 52701

ACTLDAYS: 9277
ASOFDATE: 670327
CDR: HENCK
CMSNDATE: 641107
DURATN: 4.0
ENDATE: 670705
FCLTY: NORVA
HISTDATA: OVHL
HOMEPORT: NORVA
HULL: CG 26
INTRVL: 37.0
NAVY: NAVSEA
PROC/DATE: 751114
RCDID: 527013931
SHIP: BELKNAP
STARTDATE: 670327
TYCOM: 2
TYPEOVHL: RA
UIC: 52701

THE POWER OF NAVLIS TO AUDIT

FULL DISCLOSURE ON THE BELKNAP

DEMONSTRATION OF THE INTEGRATION OF THE ARCHITECTURE

REPORT DFT QS6001 DEC 12 1975

ACTLDAYS: 45422
ASOFDATE: 680904
CDR: HENCK
CMSNDATE: 641107
DURATN: 4.0
ENDATE: 690404
FCLTY: NORVA
HISTDATA: OVHL
HOMEPORT: NORVA
HUL: CG 26
INTRVL: 37.0
NAVY: NAVSEA
PROC/DATE: 751114
RCDID: 527013932
SHIP: BELKNAP
STARTDATE: 680904
TYCOM: 2
TYPEOVHL: RO
UIC: 52701

ACTLDAYS: 62194
ASOFDATE: 720327
CDR: HENCK
CMSNDATE: 641107
DURATN: 4.0
ENDATE: 721113
FCLTY: NORVA
HISTDATA: OVHL
HOMEPORT: NORVA
HULL: CG 26
INTRVL: 37.0
NAVY: NAVSEA
PROC/DATE: 751114
RCDID: 527013933
SHIP: BELKNAP
STARTDATE: 720327
TYCOM: 2
TYPEOVHL: RO
UIC: 52701

THE POWER OF NAVLIS TO AUDIT

FULL DISCLOSURE ON THE BELKNAP

DEMONSTRATION OF THE INTEGRATION OF THE ARCHITECTURE

DATA BASE - BASE2
DFT REPORT DEFINITION NOT FOUND

REPORT DFT 0S6001 DEC 12 1975

ASOFDATE: 750731
CDR: SHARP
CNTYRPLN: 500
ENDATE: 7802
ESTCNTYRQ: 500
FCLTY: C
FNDACTVTY: 76AFR
HULL: CG 26
NAVY: CINCLANTFLT
ROC/DATE: 751114
RCDID: 527015507
SHIP: BELKNAP
STARTDATE: 7702
TYCOM: 2
UIC: 52701

ASOFDATE: 750831
CDR: SHARP
CNTYRPLN: 600
ENDATE: 7802
ESTCNTYRQ: 600
FCLTY: C
FNDACTVTY: 76AFR
HULL: CG 26
NAVY: CINCLANTFLT
PROC/DATE: 751209
RCDID: 527015663
SHIP: BELKNAP
STARTDATE: 7702
TYCOM: 2
UIC: 52701

THE POWER OF NAVLIS TO AUDIT

FULL DISCLOSURE ON THE BELKNAP

DEMONSTRATION OF THE INTEGRATION OF THE ARCHITECTURE

REPORT DFT QS6001 DEC 12 1975

ASOFDATE: 770415
ASSOC: DLG
CDR: WARD
DURATN: 9.0
HULL: CG 26
INTRVL: 37.0
NAVY: CINCLANTFLT
PROC/DATE: 751114
RCDID: 527010014
SCDACTVTY: PROJ
SHIP: BELKNAP
STARTDATE: 7512
UIC: 52701

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS
QUERY SET
IF FNDACTVTY IS 73BDGT OR 74BDGT OR 75BDGT OR 76BDGT?
REPORT S06
SORT ON FNDACTVTY ANDHULL HULL
PRINT SUBTOTAL BDGTLDAYS, BDGTLMNY, BDGTMMNY, BDGTUCOST ON FNDACTVTY
PRINT SUM ON
HEAD 1 $THE POWER OF NAVLIS TO RECONCILE$
HEAD 2 $WHAT WE SIAD WE WERE GOINT TO DO$
$END

YOUR QUER HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY SET, QUESTION NO.   1, HAS        266 HITS-T
```

APPORTIONMENT

THE POWER OF NAVLIS TO RECONCILE

WHAT WE SIAD WE WERE GOINT TO DO

REPORT SØ6 SEDØØ1 DEC 12 1975

HULL	MAN HOURS	LABOR FUND	MATL FUND	UNIT COST	RCDID
AD 38		1615	325	1940	058372598
AR 28		3516	688	4204	046462605
ARC 2		2503	496	2999	755622607
ARS 43		1049	222	1271	025382606
AS 16		2805	695	3500	046262581
AS 18		2525	448	2973	046282582
ASR 14		513	92	605	047132583
CG 10		7906	1698	9604	036232584
CG 16		2888	591	3479	526872585
CG 28		3289	589	3878	527032586
CG 32		3271	585	3856	527072587
CV 60		18277	3739	22016	033602573
DD 822		2884	515	3399	521222613
DD 841		1697	527	2224	521412588
DD 850		2606	522	3128	521502589
DD 864		2549	522	3071	521642614
DD 865		3047	545	3592	521652590
DD 871		2828	566	3394	521712591
DD 878		3047	545	3592	521782592
DD 883		2639	528	3167	521832593
DD 942		2965	538	3503	522012594
DD 944		2718	560	3278	522032595
FF 1043		2155	416	2571	540392596
FFG 4		1901	366	2267	046952597
LPD 4		3574	763	4337	071752599
LSD 32		3695	788	4483	031322600
LSD 34		3860	824	4684	031342601
MSC 198		259	43	302	164612624
MSC 199		259	43	302	164622625
MSC 205		197	41	238	164682626
MSC 206		197	41	238	164692627
MSC 207		208	41	249	164702628
MSO 427		492	81	573	079572615
MSO 429		539	89	628	079592616
LDV ERR 0003					
OVERLAY NOT FOUND					
DATA BASE - BASE2					
MSO 431		354	89	443	079612617
MSO 433		831	152	983	079632608
MSO 439		417	81	498	079692618
MSO 441		513	103	616	079712619
MSO 445		470	86	556	079752609
MSO 446		470	86	556	079762610
MSO 449		470	86	556	079792611
MSO 455		393	81	474	079852620
MSO 456		470	86	556	079862612
MSO 464		492	81	573	079942621
MSO 489		417	81	498	081472622
MSO 492		492	81	573	081502623
PG 101		381	71	452	200942604
PG 94		381	71	452	200872602

APPORTIONMENT

THE POWER OF NAVLIS TO RECONCILE

WHAT WE SIAD WE WERE GOINT TO DO

REPORT SØ6 SETØØ1 DEC 12 1975

HULL	MAN HOURS	LABOR FUND	MATL FUND	UNIT COST	RCDID
PG 99		381	71	452	200922603
SS 351		3545	609	4154	054512574
SSBN 602		29552	5609	35161	051102629
SSBN 608		26532	5609	32141	051162630
SSBN 609		25374	5408	30782	051172631
SSN 571		17316	3848	21164	055912575
SSN 597		20603	3498	24101	050602576
SSN 607		15809	3498	19307	051152577
SSN 638		11053	2456	13509	051312578
SSN 646		12065	2826	14891	051332579
SSN 649		12672	2768	15440	051362580
SUBTOTAL					
	0	275926	56507	332433	
AD 18	14341	1750	416	2166	046382650
AE 21		1753	607	2360	088212674
AE 27		1795	617	2412	058392658
AF 58	18883	2096	402	2498	015972639
AFS 6		2716	607	3323	201162659
AO 147		3483	967	4450	059072660
AO 98		4263	891	5154	048482652

Report abbreviated for this appendix.

CINCLANTFLT

Ship Funding Scenario

Terminal Session and Report Samples

Attachment #3

```
NSRDC 6700 INTERCOM V4.2
DATE  12/15/75
TIME  13.24.34.
PLEASE LOGIN
LOGIN,PUWEWEAVER,1180600823,SUP

COMMAND- ETL,500

COMMAND- COMRADE,SHARP



COMRADE         TIME:       13.25.33.
                DATE:       12/15/75

SHARP SUBSYSTEM ENTERED
???? IQUERY

PLEASE INPUT QUERY COMMANDS
QUERY 003
IF ASOFDATE IS 740731 AND SUFFIX ENDACTVTY IS AER AND PREFIX ENDATE
IS LE TO 757604?
PRINT HULL AND SHIP
HEAD 1 $SHIPS THAT END WITHIN THE FIRST 3/4S$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY 003, QUESTION NO.    1, HAS           5 HITS-T
```

SHIPS THAT END WITHIN THE FIRST 3/4S

REPORT DFT 003001 DEC 15 1975

HULL: LSD 29
SHIP: PLYMOUTH ROCK

HULL: LPD 14
SHIP: TRENTON

HULL: PG 86
SHIP: ANTELOPE

HULL: PG 87
SHIP: READY

HULL: LST 1181
SHIP: SUMTER

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?-
   LDV ERR 0003
   OVERLAY NOT FOUND
 DATA BASE - BASE2
 DFT REPORT DEFINITION NOT FOUND          NO

???? IQUERY


YOU HAVE SELECTED THE INTERACTIVE QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-NO


SELECT ONE OF THE FOLLOWING DATA BASES
18 LOGISTICS DOCUMENT DATA BANK
31 YURSO DATA BANK
32 MKRS-BASE-2
?? %A


USER ABORT

???? REPDEF


 PLEASE INPUT REPORT DEFINITIONS.

ID BASE2
REPORT 003
DBTYPE IS RCDID
MHEAD $FUNDING PROFILE REPORT$
COL HEAD 1 = $HULL$
COL HEAD 2 = $AS OF/DATE$
COL HEAD 3 = $PRIOR/YRFND/(1000$)$
COL HEAD 4 = $CNTYR/PLAN/(1000)$
COL HEAD 5 = $CNTYR/FND/(1000)$
COL HEAD 6 = $EST CNTYR/RQMT$
PRINT HULL ASOFDATE PYRYRFND, CNTYRPLN, CNTYRFND, ESTCNTYRQT
PRINT SUBTOTAL PRYRYRFND, CNTYRPLN, CNTYRFND, ESTCNTYRQT ON HULL
$END
```

```
REPORT DEFINITION ERROR SUMMARY

ID - BASE2


REPORT - 003
NO ERRORS FOUND - REPORT DEFINITION ACCEPTED
????
 EX ID=      CADC PFN=DATADEFFILE
 EX CY= 001  00001751 PRUS  $0004.38 /DAY
IQUERY


YOU HAVE SELECTED THE INTERACTIVE QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-NO


SELECT ONE OF THE FOLLOWING DATA BASES
18 LOGISTICS DOCUMENT DATA BANK
31 YURSO DATA BANK
32 MKRS-BASE-2
?? 32


PLEASE INPUT QUERY COMMANDS
QUERY 004
IF HULL IS LSD 29 OR LPD 14 OR PG 86 OR PG 87 OR LST 1181
AND ENDACTVTY
AND ENDACTVTY
AND SUFFIX ENDACTVTY IS COR OR CFY OR SRA OR AFR?
REPORT 003
HEAD 1 $AFR ASOFJULY FY75 SHIPS COMPLETING PRIOR TO 4TH QUARTER$
SORT ON HULL AND ASOFDATE
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY 004, QUESTION NO.   1, HAS        70 HITS-T
```

FUNDING PROFILE REPORT

AFR ASOFJULY FY75 SHIPS COMPLETING PRIOR TO 4TH QUARTER

REPORT 003 00400' DEC 15 1975

HULL	AS OF DATE	PRIOR YRFND (1000)	CNTYR PLAN (1000)	CNTYR FND (1000)	EST CNTYR RQMT	RCDID
LPD 14	JUL 31 74		760		760	072004345
LPD 14	AUG 31 74		760		760	072004197
LPD 14	SEP 30 74		760	7	760	072000302
LPD 14	OCT 31 74	0	760	187	760	072000471
LPD 14	NOV 30 74		380	187	760	072000647
LPD 14	DEC 31 74		380	187	760	072000848
LPD 14	JAN 31 75		380	187	380	072001049
LPD 14	FEB 28 75		550	240	550	072001316
LPD 14	MAR 31 75		550	440	550	072004899
LPD 14	APR 30 75	0	550	440	550	072005097
LPD 14	MAY 31 75		440	440	440	072004698
LPD 14	JUN 30 75		416	416	416	072005303
LPD 14	JUL 31 75	416	7344	4905	7344	072005437
LPD 14	AUG 31 75	416	6291	5405	6291	072005592
		SUBTOTAL				
		832	20321	13041	21081	
LSD 29	JUL 31 74		600		600	031294347
LSD 29	AUG 31 74		600		600	031294199
LSD 29	SEP 30 74		600	7	600	031290304
LSD 29	OCT 31 74		400	191	600	031290474
LSD 29	NOV 30 74		400	191	600	031290649
LSD 29	DEC 31 74		400	191	600	031290850
LSD 29	JAN 31 75	0	400	191	400	031291051
LSD 29	FEB 28 75		475	475	475	031291318
LSD 29	MAR 31 75		475	475	475	031294901
LSD 29	APR 30 75		515	515	515	031295098
LSD 29	MAY 31 75		530	530	530	031294700
LSD 29	JUN 30 75		530	530	530	031295305
LSD 29	JUL 31 75	530	5783	5035	5783	031295437
LSD 29	AUG 31 75	530	5460	5035	5460	031295593
		SUBTOTAL				
		1060	17168	13366	17768	
LST 1181	JUL 31 74		256		256	200204349
LST 1181	AUG 31 74		256		256	200204201
LST 1181	SEP 30 74		256		256	200200306
LST 1181	OCT 31 74		256		256	200200476
DATA BASE - BASE2						
LST 1181	NOV 30 74		256	191	256	200200651
LST 1181	DEC 31 74		256	191	650	200200852
LST 1181	JAN 31 75		356	236	356	200201053

FUNDING PROFILE REPORT

AFR ASOFJULY FY75 SHIPS COMPLETING PRIOR TO 4TH QUARTER

REPORT 003 004001 DEC 15 1975

HULL	AS OF DATE	PRIOR YREND (1000)	CNTYR PLAN (1000)	CNTYR END (1000)	EST CNTYR RQMT	RCDID
LST 1181	FEB 28 75		690	680	690	200201320
LST 1181	MAR 31 75		690	680	690	200204903
LST 1181	APR 30 75		690	680	690	200205100
LST 1181	MAY 31 75		680	680	680	200204702
LST 1181	JUN 30 75		680	680	680	200205307
LST 1181	JUL 31 75	680	4430	2950	4430	200205438
LST 1181	AUG 31 75	680	3851	3797	3851	200205594
	SUBTOTAL					
		1360	13603	10765	13997	
PG 86	JUL 31 74		28		28	125854342
PG 86	AUG 31 74		28		28	125854191
PG 86	SEP 30 74		28		28	125850296
PG 86	OCT 31 74		28		28	125850465
PG 86	NOV 30 74		28		28	125850640
PG 86	DEC 31 74			28	28	125850842
PG 86	JAN 31 75		28		28	125851043
PG 86	FEB 28 75		28		28	125851313
PG 86	MAR 31 75		28		28	125854896
PG 86	APR 30 75		28	28	28	125855094
PG 86	MAY 31 75		28	28	28	125854695
PG 86	JUN 30 75		30	30	30	125855297
PG 86	JUL 31 75	30	482	100	482	125855451
PG 86	AUG 31 75	30	482	125	482	125855607
	SUBTOTAL					
		60	1274	339	1302	
PG 87	JUL 31 74		27		27	125864343
PG 87	AUG 31 74		27		27	125864192
PG 87	SEP 30 74		27		27	125860297
PG 87	OCT 31 74		27		27	125860466
PG 87	NOV 30 74		27		27	125860641
PG 87	DEC 31 74		27		27	125860843
PG 87	JAN 31 75		27		27	125861044
PG 87	FEB 28 75		27		27	125861314
PG 87	MAR 31 75		27		27	125864897
PG 87	APR 30 75		27	27	27	125865095
PG 87	MAY 31 75		27	27	27	125864696
PG 87	JUN 30 75		29	29	29	125865298
PG 87	JUL 31 75	29	485	150	485	125865452
PG 87	AUG 31 75	29	485	175	485	125865608

FUNDING PROFILE REPORT

AFR ASOFJULY FY75 SHIPS COMPLETING PRIOR TO 4TH QUARTER

REPORT 003 004001 DEC 15 1975

HULL	AS OF DATE	PRIOR YREND (1000)	CNTYR PLAN (1000)	CNTYR FND (1000)	EST CNTYR RQMT	RCDID
		SUBTOTAL				
		58	1296	408	1296	

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES


PLEASE INPUT QUERY COMMANDS
QUERY ØØ5
IF FNDACTVTY IS 74BDGT AND HULL IS LPD 14 OR LSD 29 OR LST 1181
OR PG 86 OR PG 87?
REPORT SØ6
SORT ON HULL
HEAD 1 $74 BDGT INFORMATION$
IF FNDACTVTY IS 75BDGT AND HULL IS PLPD 14 OR LSD 29 OR LST 1181
OR PG 86 OR PG 87?
REPORT SØ6
SORT ON HULL
HEAD 1 $75 BDGT INFORMATION$
IF FNDACTVTY IS 76BDGT AND HULL IS LPD 14 OR LSD 29 OR LST 1181
OR PG 86 OR PG 87?
REPORT SØ6
SORT ON HULL
HEAD 1 $76 BDGT INFORMATION$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY ØØ5, QUESTION NO.   1, HAS         2 HITS-T

QUERY ØØ5, QUESTION NO.   2, HAS         Ø HITS
QUERY ØØ5, QUESTION NO.   3, HAS         5 HITS-T
```

APPORTIONMENT

74 BDGT INFORMATION

REPORT S06 005001 DEC 15 1975

HULL	MAN HOURS	LABOR FUND	MATL FUND	UNIT COST	RCDID
PG 86		485	120	605	125852656
PG 87		460	113	573	125862657

APPORTIONMENT

76 BDGT INFORMATION

REPORT S06 005003 DEC 15 1975

HULL	MAN HOURS	LABOR FUND	MATL FUND	UNIT COST	RCDID
LPD 14		6344	1440	7784	072005644
LSD 29		5064	1249	6313	031295645
LST 1181		4002	1108	5110	200205647
PG 86		443	67	510	125855652
PG 87		445	67	512	125865653

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- NO

???? REPDEF


 PLEASE INPUT REPORT DEFINITIONS.

ID BASE2
REPORT ØØ5
DBTYPE IS RCDID
MHEAD $APPORTIONMENT INFORMATION$
COL HEAD 1 = $FND/ACTVTY$
COL HEAD 2 = $HULL$
COL HEAD 3 = $MAN/DAYS$
COL HEAD 4 = $LABOR/FUND/(1ØØØ)$
COL HEAD 5 = $MATL/FNDING/(1ØØØ)$
COL HEAD 6 = $UNIT/COST/(1ØØØ)$
PRINT FNDACTVTY, HULL, BDGTLDAYS, BDGTLMNY,BDGTMMNY, BDGTUCOST
$END

REPORT DEFINITION ERROR SUMMARY

ID - BASE2



REPORT - ØØ5
NO ERRORS FOUND - REPORT DEFINITION ACCEPTED
???? IQUERY



YOU HAVE SELECTED THE INTERACTIVE QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-NO


SELECT ONE OF THE FOLLOWING DATA BASES
18 LOGISTICS DOCUMENT DATA BANK
31 YURSO DATA BANK
32 MKRS-BASE-2
?? 32

PLEASE INPUT QUERY COMMANDS
QUERY ØØ6
IF SUFFIX FNDACTVTY IS BDGT AND HULL IS LPD 14 OR LSD 29 OR LST 1181
OR PG 86 OR PG 87?
REPORT ØØ5
SORT ON FNDACTVTY AND HULL
HEAD 1 $LPD 14 - LSD 29 - LST 1181 - PG 86 - PG 87 HULLS$
HEAD 2 $DEMONSTRATION OF THE INTEGRATED QUERY$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR R.

QUERY ØØ6, QUESTION NO.    1, HAS           7 HITS-T
```

APPORTIONMENT INFORMATION

FIVE SHIP SCENARIO

DEMONSTRATION OF THE INTEGRATED QUERY

REPORT 005 006001 DEC 15 1975

FND ACTVTY	HULL	MAN DAYS	LABOR FUND (1000)	MATL FNDING (1000)	UNIT COST (1000)	RCDID
74BDGT	PG 86		485	120	605	125852656
74BDGT	PG 87		460	113	573	125862657
76BDGT	LPD 14		6344	1440	7784	072005644
76BDGT	LSD 29		5064	1249	6313	031295645
76BDGT	LST 1181		4002	1108	5110	200205647
76BDGT	PG 86		443	67	510	125855652
76BDGT	PG 87		445	67	512	125865653

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?-
 DATA BASE - BASE2          YES

PLEASE INPUT QUERY COMMANDS
QUERY 007
IF HULL IS LPD 14 OR LSD 29 OR LST 1181 OR PG 86 OR PG 87?
REPORT PSB
SORT ON HULL AND ASOFDATE
HEAD 1 $INFORMATION ABOUT INFORMATION IN THE BASE$
HEAD 2 $FIVE SHIP RECORD POPULATION ANANYSIS$
HEAD 3 $NAVLIS IS OPERATIONAL$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR P.

QUERY 007, QUESTION NO.   1, HAS        90 HITS-T
```

SHIP OVERHAUL PROFILE

INFORMATION ABOUT INFORMATION IN THE BASE

FIVE SHIP RECORD POPULATION ANANYSIS

NAVLIS IS OPERATIONAL

REPORT PSB 007001 DEC 15 1975

ASOF DATE	START DATE	END DATE	TYPE OVHL	END ACTVTY	SCHED ACTVTY	RCDID
NOV 09 71	NOV 09 71	DEC 20 71	PS			072003287
JUL 31 74	AUG 75	APR 76		75AFR		072004345
AUG 31 74	AUG 75	APR 76		75AFR		072004197
SEP 30 74	AUG 75	APR 76		75AFR		072000302
OCT 31 74	AUG 75	APR 76		75AFR		072000471
NOV 30 74	AUG 75	APR 76		75AFR		072000647
DEC 31 74	AUG 75	APR 76		75AFR		072000848
JAN 31 75	AUG 75	APR 76		75AFR		072001049
FEB 28 75	AUG 75	APR 76		75AFR		072001316
MAR 31 75	AUG 75	APR 76		75AFR		072004899
APR 30 75	AUG 75	APR 76		75AFR		072005097
MAY 31 75	AUG 75	APR 76		75AFR		072004698
JUN 01 75	DEC 71			76BDGT		072005644
JUN 30 75	AUG 75	APR 76		75AFR		072005303
JUL 31 75	AUG 75	APR 76		76CFY		072005437
AUG 31 75	AUG 75	APR 76		76CFY		072005592
APR 15 77	AUG 79				PROJ	072000153
APR 28 64	APR 28 64	JUL 30 64	RO			031293301
FEB 26 68	FEB 26 68	JUL 22 68	FO			031293302
JUL 01 71	JUL 01 71	NOV 01 71	RO			031293303
JUL 31 74	AUG 75	JAN 76		75AFR		031294347
AUG 31 74	AUG 75	JAN 76		75AFR		031294199
SEP 30 74	AUG 75	JAN 76		75AFR		031290304
OCT 31 74	AUG 75	JAN 76		75AFR		031290474
NOV 30 74	AUG 75	JAN 76		75AFR		031290649
DEC 31 74	AUG 75	JAN 76		75AFR		031290850
JAN 31 75	AUG 75	JAN 76		75AFR		031291051
FEB 28 75	AUG 75	JAN 76		75AFR		031291318
MAR 31 75	AUG 75	JAN 76		75AFR		031294901
APR 30 75	AUG 75	JAN 76		75AFR		031295098
MAY 31 75	AUG 75	JAN 76		75AFR		031294700
JUN 01 75	JUL 71	NOV 71		76BDGT		031295645
JUN 30 75	AUG 75	JAN 76		75AFR		031295305
JUL 31 75	AUG 75	JAN 76		76CFY		031295437
AUG 31 75	AUG 75	JAN 76		76CFY		031295593
APR 15 77	FEB 80				PROJ	031290177
DATA BASE - BASE2						
OCT 03 66	OCT 03 66	AUG 20 70	NC			200203318
JUL 13 70	JUL 13 70	AUG 20 70	FO			200203319
JUL 31 74	AUG 75	FEB 76		75AFR		200204349
AUG 31 74	AUG 75	FEB 76		75AFR		200204201
SEP 30 74	AUG 75	FEB 76		75AFR		200200306
OCT 31 74	AUG 75	FEB 76		75AFR		200200476
NOV 30 74	AUG 75	FEB 76		75AFR		200200651
DEC 31 74	AUG 75	FEB 76		75AFR		200200852
JAN 31 75	AUG 75	FEB 76		75AFR		200201053
FEB 28 75	AUG 75	FEB 76		75AFR		200201320

SHIP OVERHAUL PROFILE

INFORMATION ABOUT INFORMATION IN THE BASE

FIVE SHIP RECORD POPULATION ANANYSIS

NAVLIS IS OPERATIONAL

REPORT PSB ØØ7ØØ1 DEC 15 1975

ASOF DATE	START DATE	END DATE	TYPE OVHL	END ACTVTY	SCHED ACTVTY	RCDID
MAR 31 75	AUG 75	FEB 76		75AFR		2ØØ2Ø49Ø3
APR 3Ø 75	AUG 75	FEB 76		75AFR		2ØØ2Ø51ØØ
MAY 31 75	AUG 75	FEB 76		75AFR		2ØØ2Ø47Ø2
JUN Ø1 75	MAR 71			76BDGT		2ØØ2Ø5647
JUN 3Ø 75	AUG 75	FEB 76		75AFR		2ØØ2Ø53Ø7
JUL 31 75	AUG 75	MAR 76		76CFY		2ØØ2Ø5438
AUG 31 75	AUG 75	MAR 76		76CFY		2ØØ2Ø5594
APR 15 77	OCT 79				PROJ	2ØØ2ØØ17Ø
MAY Ø1 73	JUN 71	JAN 72		74BDGT		125852656
MAR Ø1 74	MAR Ø1 74	JUN 11 74	RO			125854Ø14
JUL 31 74	JAN 76	APR 76		75AFR		125854342
AUG 31 74	JAN 76	APR 76		75AFR		125854191
SEP 3Ø 74	JAN 76	APR 76		75AFR		12585Ø296
OCT 31 74	JAN 76	APR 76		75AFR		12585Ø465
NOV 3Ø 74	JAN 76	APR 76		75AFR		12585Ø64Ø
DEC 31 74	JAN 76	APR 76		75AFR		12585Ø842
JAN 31 75	JAN 76	APR 76		75AFR		1258510Ø43
FEB 28 75	JAN 76	APR 76		75AFR		125851313
MAR 31 75	JAN 76	APR 76		75AFR		125854896
APR 3Ø 75	JAN 76	APR 76		75AFR		1258550Ø94
MAY 31 75	JAN 76	APR 76		75AFR		125854695
JUN Ø1 75	MAR 74	MAY 74		76BDGT		125855652
JUN 3Ø 75	JAN 76	APR 76		75AFR		125855297
JUL 31 75	JAN 76	APR 76		76CFY		125855451
AUG 31 75	JAN 76	APR 76		76CFY		1258556Ø7
APR 15 77	AUG 78				PROJ	12585ØØ99
MAY Ø1 73	JUN 71	JAN 72		74BDGT		125862657
MAR Ø1 74	MAR Ø1 74	JUN 11 74	RO			1258640Ø15
JUL 31 74	JAN 76	APR 76		75AFR		125864343
AUG 31 74	JAN 76	APR 76		75AFR		125864192
SEP 3Ø 74	JAN 76	APR 76		75AFR		12586Ø297
OCT 31 74	JAN 76	APR 76		75AFR		12586Ø466
NOV 3Ø 74	JAN 76	APR 76		75AFR		12586Ø641
DEC 31 74	JAN 76	APR 76		75AFR		12586Ø843
JAN 31 75	JAN 76	APR 76		75AFR		125861Ø44
FEB 28 75	JAN 76	APR 76		75AFR		125861314
MAR 31 75	JAN 76	APR 76		75AFR		125864897
APR 3Ø 75	JAN 76	APR 76		75AFR		125865Ø95
MAY 31 75	JAN 76	APR 76		75AFR		125864696
JUN Ø1 75	MAR 74	MAY 74		76BDGT		125865653
JUN 3Ø 75	JAN 76	APR 76		75AFR		125865298
JUL 31 75	JAN 76	APR 76		76CFY		125865452
AUG 31 75	JAN 76	APR 76		76CFY		1258656Ø8
APR 15 77	AUG 78				PROJ	12586Ø1ØØ

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS
QUERY 008
IF RCDID IS 072003287 OR 031293301 OR 200203318 OR 125852656
OR 125862657?
PRINT RCDID, HULL, SHIP, UIC
SORT ON RCDID
HEAD 1 $DEMONSTRATION OF THE MULTIPHASE SPLIT KEY RETRIEVAL POWER$
HEAD 2 $END OF THE FIVE SHIP SCENARIO$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY 008, QUESTION NO.   1, HAS          5 HITS-
ILLEGAL CHARACTER, TRY AGAIN T
```

DEMONSTRATION OF THE MULTIPHASE SPLIT KEY RETRIEVAL POWER

REPORT DFT 008001 DEC 15 1975

RCDID: 031293301
HULL: LSD 29
SHIP: PLYMOUTH ROCK
UIC: 3129

RCDID: 072003287
HULL: LPD 14
SHIP: TRENTON
UIC: 7200

RCDID: 125852656
HULL: PG 86
SHIP: ANTELOPE
UIC: 12585

RCDID: 125862657
HULL: PG 87
SHIP: READY
UIC: 12586

RCDID: 200203318
HULL: LST 1181
SHIP: SUMTER
UIC: 20020

AD-A035 847 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 9/2
NAVY LOGISTICS INFORMATION SHARING (NAVLIS) PROJECT.(U)
JUN 76

UNCLASSIFIED DTNSRDC-76-0120 NL

5 OF 5
AD
A035847

END
DATE
FILMED
3-77

CINCLANTFLT

Scenario of August 1975

Funding Report

Terminal Session and Report Samples

Attachment #4

```
NSRDC 6700 INTFRCOM V4.2
DATE  12/16/75.
TIME  13:18:28.

PLFASF LOGIN
LOGIN,PUWFWFAVFR,1180600823,SUP

COMMAND- FTL,500

COMMAND- COMRADF,SHARP


COMRADF        TIMF:      13.19.33.
               DATF:      12/16/75

SHARP SUBSYSTFM FNTFRFD
???? RFPDFF

 PLFASF INPUT RFPORT DFFINITIONS.

ID BASF2
RFPORT FND
MHFAD $SHIP FUNDING ACTVITY$
COL HFAD 1 = $HULL$
COL HFAD 2 = $FND/ACTVTY$
COL HFAD 3 = $PYRYR/FND/(1000)$
COL HFAD 4 = $CNTYR/PLAN/(1000)$
COL HFAD 5 = $CNTPLN/ALLYRS/(1000)$
COL HFAD 6 = $TOTFND/ALLYRS/TODTF$
COL HFAD 7 = $CNTYR/FND/TODTF$
COL HFAD 8 = $FST/CNTYR/RQMT$
COL HFAD 9 = $FST/THOR/OVHL$
PRINT HULL, PYRYFND, CNTYRPLN, CNTPLNTOT, TOTFND, CNTYRFND,
ESTCNTYRQT, ESTOVHL
$FND

RFPORT DFFINITION FRROR SUMMARY

ID - BASF2



RFPORT - FND
NO FRRORS FOUND - RFPORT DFFINITION ACCEPTFD
???? RFPDFF


 PLFASF INPUT RFPORT DFFINITIONS.

ID BASF2
RFPORT FLT
MHFAD $SHIP SCHFDULF RFPORT$
COL HFAD 1 = $FND/DATF$
COL HFAD 2 = $START/DATF$
COL HFAD 3 = $HULL$
COL HFAD 4 = $SHIPS$
COL HFAD 5 = $YARD$.
COL HFAD 6 = $%/OVHL$
PRINT FNDATF, STARTDATF, HULL, SHIP, FCLTY, PFRCNTOVHL
$FND
```

```
REPORT DEFINITION ERROR SUMMARY

ID - BASE2


REPORT - FLT
NO ERRORS FOUND - REPORT DEFINITION ACCEPTED
???? IQUERY


YOU HAVE SELECTED THE INTERACTIVE QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-NO


SELECT ONE OF THE FOLLOWING DATA BASES
18 LOGISTICS DOCUMENT DATA BANK
31 YURSO DATA BANK
32 MKRS-BASE-2
?? 32

PLEASE INPUT QUERY COMMANDS
QUERY S003
QUERY S03
IF ASOFDATE IS 750831?
REPORT SCD
SORT ON (*5,4)RCDID
HEAD 1 $AUGUST 1975 INFORMATION$
HEAD 2 $ATLANTIC FLEET ACTIVITY$
HEAD 3 $FUNDING SCENARIO$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY S03, QUESTION NO.   1, HAS        166 HITS-T
```

SHIP FUNDING SCHEDULE

AUGUST 1975 INFORMATION

ATLANTIC FLEET ACTIVITY

FUNDING SCENARIO

REPORT SCD S03001 DEC 16 1975

HULL	SHIP	YARD	START DATE	END DATE	END ACTVTY
SSN 678	ARCHERFISH		JUN 76	JUN 77	76CFY
SSN 585	SKIPJACK	SG	JUL 74	JUL 76	76COR
SSN 591	SHARK	SP	AUG 74	FEB 76	76COR
SSN 604	HADDO	SP	AUG 73	JUL 75	76COR
SSN 664	SEA DEVIL	PT	JUL 74	JUN 75	76COR
SSN 670	FINBACK	N	JUL 74	MAY 75	76COR
SSN 674	TREPANG	PT	OCT 74	OCT 75	76COR
SSN 676	BILLFISH	PT	JAN 75	JAN 76	76COR
CV 66	AMERICA	N	DEC 74	OCT 75	76COR
AFS 2	SYLVANIA	5	MAY 75	NOV 75	76COR
AO 99	CANISTEO	3	MAR 75	SEP 75	76COR
ARS 6	ESCAPE	SJ	JUN 75	NOV 75	76COR
CG 34	BIDDLE	BA	APR 75	APR 76	76COR
DD 941	DU PONT	N	APR 75	DEC 75	76COR
DDG 3	KING	N	MAY 75	MAR 76	76COR
FF 1059	SIMS	PH	JUN 75	MAR 77	76COR
FFG 6	FURER	C	JAN 75	OCT 75	76COR
LPD 1	RALEIGH	5	AUG 74	SEP 75	76COR
LSD 30	FORT SNELLING	5	FEB 75	OCT 75	76COR
LSD 37	PORTLAND	5	JUN 75	DEC 75	76COR
PG 93	WELCH	5	JUN 75	OCT 75	76COR
AG 520	ALACRITY	6	JUN 75	OCT 75	76COR
AG 521	ASSURANCE	6	JUN 75	OCT 75	76COR
FF 1038	MCCLOY	PH	MAR 75	SEP 75	76COR
AD 38	PUGET SOUND	N	MAY 75	SEP 75	75COR
DDG 45	DEWEY	C	APR 75	MAY 76	76COR
FFG 5	PAGE	PH	APR 75	JAN 76	76COR
DDG 2	ADAMS	PH	FEB 75	DEC 75	76COR
AOR 2	MILWAUKEE	PH	FEB 75	SEP 75	76COR
LST 1180	MANITOWOC	5	JAN 75	AUG 75	76COR
DD 937	DAVIS	C	JAN 75	DEC 75	76COR
DD 938	INGRAM	C	OCT 75	SEP 75	76COR
DDG 17	CONYNGHAM	N	OCT 75	AUG 75	76COR
FF 1072	BLAKELY	C	JUL 75	MAR 76	76CFY
LPH 9	GUAM	PH	JUL 75	FEB 76	76CFY
SSN 675	BLUEFISH	N	JUL 75	JUL 76	76CFY
DDG 23	BYRD	N	AUG 75	JUN 76	76CFY
DATA BASE - BASE2					
LPD 14	TRENTON	5	AUG 75	APR 76	76CFY
LSD 29	PLYMOUTH ROCK	5	AUG 75	JAN 76	76CFY
LST 1181	SUMTER	5	AUG 75	MAR 76	76CFY
SSN 653	RAY	C	AUG 75	FEB 77	76CFY
SSN 606	TINOSA	SP	AUG 75	FEB 77	76CFY
SSN 605	JACK	PT	OCT 75	FEB 77	76CFY
DDG 19	TATTNALL	C	OCT 75	AUG 76	76CFY
FF 1075	TRIPPE	C	OCT 75	AUG 76	76CFY
DD 943	BLANDY	3	OCT 75	SEP 76	76CFY

SHIP FUNDING SCHEDULE

AUGUST 1975 INFORMATION

ATLANTIC FLEET ACTIVITY

FUNDING SCENARIO

REPORT SCD S03001 DEC 16 1975

HULL	SHIP	YARD	START DATE	END DATE	END ACTVTY
ATF 162	SHAKORI	5	NOV 75	MAR 76	76CFY
DDG 10	SAMPSON	PH	NOV 75	OCT 76	76CFY
FF 1068	VREELAND	PH	DEC 75	NOV 76	76CFY
DDG 4	LAWRENCE	N	DEC 75	NOV 76	76CFY
AE 28	SANTA BARBARA	SJ	DEC 75	SEP 76	76CFY
AD 19	YOSEMITE	SJ	JAN 76	JUN 76	76CFY
PG 86	ANTELOPE	NELM	JAN 76	APR 76	76CFY
PG 87	READY	NELM	JAN 76	APR 76	76CFY
SSN 650	PARGO	SG	JAN 76	APR 77	76CFY
AS 34	CANOPUS	C	JAN 76	SEP 76	76CFY
SSN 660	SAND LANCE	C	JAN 76	JAN 77	76CFY
AR 5	VULCAN	5	JAN 76	AUG 76	76CFY
DD 940	MANLEY	PH	JAN 76	DEC 76	76CFY
DD 933	BARRY	PH	JAN 76	DEC 76	76CFY
SSBN 629	BOONE	PT	JAN 76	MAY 77	76CFY
LSD 38	PENSACOLA	5	FEB 76	AUG 76	76CFY
LST 1188	SAGINAW	5	FEB 76	AUG 76	76CFY
AGF 3	LA SALLE	SUBIC	MAR 76	JUN 76	76CFY
SSN 661	LAPON	NN	MAR 76	JUN 76	76CFY
LPD 12	SHREVEPORT	5	MAR 76	OCT 76	76CFY
DD 931	SHERMAN	3	MAR 76	JAN 77	76CFY
AS 36	SPEAR	N	APR 76	OCT 76	76CFY
AOE 4	DETROIT	5	APR 76	FEB 77	76CFY
FF 1078	HEWES	C	APR 76	FEB 77	76CFY
FF 1079	BOWEN		MAY 76	MAR 77	76CFY
LPH 12	INCHON		MAY 76	NOV 76	76CFY
DDG 38	LUCE		JUN 76	JUL 77	76CFY
MSO 448	ILLUSIVE		JUN 76	SEP 76	76CFY
CV 42	ROOSEVELT	SJ	SEP 75	DEC 75	76SRA

Report abbreviated for this appendix.

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?-
NO


???? REPDEF


KERR IS =          U
IERR IS =00000000000000
BIN-ZERO=

DELAY IN OBTAINING REPORT DEFINITION FILE
ARE YOU WILLING TO WAIT A FEW SECONDS?-YES

KERR IS =          U
IERR IS =00000000000000
BIN-ZERO=

DELAY IN OBTAINING REPORT DEFINITION FILE
ARE YOU WILLING TO WAIT A FEW SECONDS?-YES


 PLEASE INPUT REPORT DEFINITIONS.

ID BASE2
REPORT FND
MHEAD $SHIP FUNDING ACTIVITY$
COL HEAD 1 = $HULLS
COL HEAD 2 = $FND/ACTVTY$
COL HEAD 3 = $PYRYR/FND/(1000)$
COL HEAD 4 = $CNTYR/PLAN/(1000)$
COL HEAD 5 = $CNTPLN/ALLYRS/(1000)$
COL HEAD 6 = $TOTFND/ALLYRS/TODTE$
COL HEAD 7 = $CNTYR/FND/TODTE$
COL HEAD 8 = $EST/CNTYR/RQMT$
COL HEAD 9 = $EST/THOR/OVHL$
PRINT HULL, FNDACTVTY, PYRYRFND, CNTYRPLN, CNTPLNTOT, TOTFND,
CNTYRFND, ESTCNTYRQT, ESTOVHL
$END


REPORT DEFINITION ERROR SUMMARY

ID - BASE2



REPORT - FND
NO ERRORS FOUND - REPORT DEFINITION ACCEPTED
???? IQUERY
```

```
YOU HAVE SELECTED THE INTERACTIVE QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-NO


SELECT ONE OF THE FOLLOWING DATA BASES
18 LOGISTICS DOCUMENT DATA BANK
31 YURSO DATA BANK
32 MKRS-BASE-2
?? 32


PLEASE INPUT QUERY COMMANDS
QUERY S03
IF ASOFDATE IS 750831?
REPORT END
SORT ON ENDACTVTY AND HULL
PRINT SUBTOTAL PYRYREND, CNTYRPLN.,CNTPLNTOT, TOTEND, CNTYREND,
ESTCNTYRQT, ESTOVHL ON ENDACTVTY
PRINT TOTAL PYRYREND, CNTYRPLN, CNTPLNTOT, TOTEND, CNTYREND,
ESTCNTYRQT, ESTOVHL
HEAD 1 $PERIOD ENDING AUGUST 17975$
HEAD 2 $ATLANTIC FLEET$
HEAD 3 $FUNDING SCENARIO$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY S03, QUESTION NO.    1, HAS        166 HITS-T

 DATA BASE - BASE2
```

SHIP FUNDING ACTIVITY

PERIOD ENDING AUGUST 1975

ATLANTIC FLEET

FUNDING SCENARIO

REPORT FND S03001 DEC 16 1975

HULL	FND ACTVTY	PYRYR FND (1000)	CNTYR PLAN (1000)	CNTPLN ALLYRS (1000)	TOTFND ALLYRS TODTE	CNTYR FND TODTE	EST CNTYR RQMT	EST THOR OVHL
AD 38	75COR	831		831	831			831
SUBTOTAL								
		831	0	831	831	0	0	831
AE 23	76AFR		700		310	310	700	
AOR 4	76AFR	87	500		382	295	500	
ARS 41	76AFR		150		91	91	150	
AS 11	76AFR		500				500	
AS 33	76AFR		500				500	
ATF 159	76AFF		120		91	91	120	
ATF 160	76AFR		60				60	
ATF 161	76AFR	1029	60		1029		60	
ATS 1	76AFR	100	437		337	237	437	
CG 19	76AFR		500		291	291	500	
CG 20	76AFR		200				200	
CG 26	76AFR		600				600	
CG 4	76AFR	210	700		286	76	700	
CV 59	76AFR	100	3000		3100	3000	3000	
DD 944	76AFR		200				200	
DDG 35	76AFR		300				300	
DDG 40	76AFR		200				200	
FF 1040	76AFR		100				100	
FF 1043	76AFR		150				150	
FF 1044	76AFR	1	500		254	253	500	
FF 1061	76AFR	1	500		288	287	500	
FF 1080	76AFR	19	600		244	225	600	
FF 1081	76AFR	100	500		280	180	500	
FF 1082	76AFR		200				200	
FF 1084	76AFR		200				200	
FF 1085	76AFR		100				100	
FF 1089	76AFR		100				100	
FFG 4	76AFR		185				185	
LPD 15	76AFR		500		50	50	500	
LPD 4	76AFR		500		150	150	500	
LPH 2	76AFR		825		150	150	825	
LSD 32	76AFR	76	600		200	124	600	
LSD 34	76AFR		400		50	50	400	
LST 1190	76AFR		750		235	235	750	
LST 1192	76AFR		600		185	185	600	
LST 1193	76AFR		300		85	85	300	
LST 1194	76AFR		400		85	85	400	

SHIP FUNDING ACTIVITY

PERIOD ENDING AUGUST 1975

ATLANTIC FLEET

FUNDING SCENARIO

REPORT FND S03001 DEC 16 1975

HULL	FND. ACTVTY	PYRYR FND (1000)	CNTYR PLAN (1000)	CNTPLN ALLYRS (1000)	TOTFND ALLYRS TODTE	CNTYR FND TODTE	EST CNTYR RQMT	EST THOR OVHL
MSO 443	76AFR		39		39	39	39	
MSO 490	76AFR		39		39	39	39	
PG 100	76AFR		25		10	10	25	
PG 98	76AFR		25		10	10	25	
SSBN 628	76AFR		4364		1565	1565	4364	
SSBN 630	76AFR		3016		162	162	3016	
SSBN 631	76AFR		3066		3015	3015	3066	
SSBN 634	76AFR		3517		1573	1573	3517	
SSBN 635	76AFR		3762		1359	1359	3762	
SSBN 636	76AFR		3569		1351	1351	3569	
SSN 607	76AFR		4		4	4	4	
SSN 614	76AFR		4550		3558	3558	4550	
SSN 615	76AFR		3550		2607	2607	3550	
SSN 637	76AFR		2232		8	8	2232	
SSN 638	76AFR		3398		106	106	3398	
SSN 646	76AFR		9		8	8	9	
SSN 649	76AFR		2281		1132	1132	2281	
SSN 663	76AFR		2880		138	138	2880	
SSN 668	76AFR		912		912	912	912	
SSN 669	76AFR		1206		1206	1206	1206	
SSN 679	76AFR		1718		107	107	1718	
SUBTOTAL								
		1723	60899	0	27082	25359	60899	0
DD 863	76AFRR		130	130	130	130	130	
DD 871	76AFRR	95	125	220	220	125	125	
DD 883	76AFRR	91	69	160	91		69	
MSO 433	76AFRR		39	39	39	39	39	
MSO 440	76AFRR		39	39	39	39	39	
MSO 509	76AFRR		39	39	39	39	39	
MSO 511	76AFRR		39	39	39	39	39	
PG 84	76AFRR		10	10	10	10	10	
PG 88	76AFRR		10	10	10	10	10	
SUBTOTAL								
		186	500	686	617	431	500	0

SHIP FUNDING ACTIVITY

PERIOD ENDING AUGUST 1975

ATLANTIC FLEET

FUNDING SCENARIO

REPORT FND S03001 DEC 16 1975

HULL	FND ACTVTY	PYRYR FND (1000)	CNTYR PLAN (1000)	CNTPLN ALLYRS (1000)	TOTFND ALLYRS TODTE	CNTYR FND TODTE	FST CNTYR RQMT	FST THOR OVHL
AD 19	76CFY	63	4000	4063	258	195	4000	4063
AE 28	76CFY	205	5000	5205	548	343	5000	5205
AGF 3	76CFY	295	3489	3784	375	80	3489	3784
AOE 4	76CFY	408	9246	9654	731	323	9246	9654
AR 5	76CFY	53	3203	3256	212	159	3203	3256
AS 34	76CFY	500	13202	13702	2000	1500	13202	13702
AS 36	76CFY	125	10596	10721	125		10596	10721
ATF 162	76CFY	88	1193	1281	830	742	1193	1281
DD 931	76CFY	20	9227	9247	355	335	9227	9247
DD 933	76CFY	216	11136	11352	900	684	11136	11352
DD 940	76CFY	193	10231	10424	923	730	10231	10424
DD 943	76CFY	481	9100	9581	521	40	9100	9581
DDG 10	76CFY	436	10876	11312	9349	8913	10876	11312
DDG 19	76CFY	1341	11500	12841	1341		11500	12841
DDG 23	76CFY	1313	1008	11394	1322	9	10081	11394
DDG 38	76CFY	25	487	512	307	282	487	512
DDG 4	76CFY	479	11293	11772	483	4	11293	11772
FF 1068	76CFY	372	8000	8372	378	6	8000	8372
FF 1072	76CFY	650	6650	7300	5763	5113	6650	7300
FF 1075	76CFY	516	6600	7116	6067	5551	6600	7116
FF 1078	76CFY	335	8417	8752	353	18	8417	8752
FF 1079	76CFY	283	8299	8582	337	54	8299	8582
LPD 12	76CFY	275	7823	8098	283	8	7823	8093
LPD 14	76CFY	416	6291	6707	5821	5405	6291	6707
LPH 12	76CFY	208	10432	10640	558	350	10432	10640
LPH 9	76CFY	2058	7770	9828	8171	6113	7770	9828
LSD 29	76CFY	530	5460	5990	5565	5035	5460	5990
LSD 38	76CFY	200	7100	7300	200		7100	7300
LST 1181	76CFY	680	3851	4531	4477	3797	3851	4531
LST 1188	76CFY	250	4930	5180	730	480	4930	5180
MSO 448	76CFY	500	500	1000	539	39	500	1000
PG 86	76CFY	30	482	512	155	125	482	512
PG 87	76CFY	29	485	514	204	175	485	514
SSBN 629	76CFY	2689	35274	37963	7696	5007	35274	37963
SSN 605	76CFY	1307	30084	31391	5315	4008	30084	31391
SSN 606	76CFY	3358	29200	32558	31061	27703	29200	32558
SSN 650	76CFY	3787	23690	27477	5096	1309	23690	27477
SSN 653	76CFY	5587	32990	38577	11593	6006	32990	38577
SSN 660	76CFY	2411	23744	26155	4918	2507	23744	26155
SSN 661	76CFY	1580	29164	30744	1787	.207	29164	30744
SSN 675	76CFY	2713	19236	21949	12717	10004	19236	21949
SSN 678	76CFY	929	2100	3029	936	7	2100	3029

SHIP FUNDING ACTIVITY

PERIOD ENDING AUGUST 1975

ATLANTIC FLEET

FUNDING SCENARIO

REPORT FND S03001 DEC 16 1975

HULL	FND ACTVTY	PYRYR FND (1000)	CNTYR PLAN (1000)	CNTPLN ALLYRS (1000)	TOTFND ALLYRS TODTE	CNTYR FND TODTE	EST CNTYR RQMT	EST THOR OVHL
				SUBTOTAL				
		37934	443359	490366	141300	103366	452432	490361
DD 821	76CFYR	75	5257	5440	4610	4427	5257	5440
DD 822	76CFYR	90	4579	4669	90		4579	4669
DD 842	76CFYR	88	5295	5383	130	42	5295	5383
DD 866	76CFYR	1	4386	4502	166	50	4386	4502
DD 880	76CFYR	50	5307	5378	3008	2937	5307	5378
LPA 249	76CFYR	538	6505	7043	738	200	6505	7964
MSO 431	76CFYR	35	586	621	555	520	586	621
MSO 441	76CFYR		581	581	39	39	581	581
MSO 442	76CFYR		496	496	39	39	496	496
MSO 446	76CFYR	5	560	565	438	433	560	565
MSO 456	76CFYR	10	636	646	360	350	636	646
PG 89	76CFYR		632	632	25	25	632	632
				SUBTOTAL				
		892	34820	35956	10198	9062	34820	36877
AFS 2	76COR	3287	500	3787	3646	359	500	3787
AG 520	76COR	630	43	673	643	13	43	673
AG 521	76COR	550	30	580	550		30	580
AO 99	76COR	4199	45	4244	4244	45	45	4244
AOR 2	76COR	6348		6348	6348			6348
ARS 6	76COR	709	95	804	774	65	95	804
CG 34	76COR	7671	400	8071	7755	84	400	8071
CV 66	76COR	30970	53	31023	31023	53	53	31023
DD 937	76COR	7658	120	7778	7658		120	7778
DD 938	76COR	7822	47	7869	7868	46	47	7869
DD 941	76COR	6940	150	7090	7004	64	150	7090
DDG 17	76COR	8068	20	8088	8087	19	20	8088
DDG 2	76COR	7567	400	7967	7567		400	7967
DDG 3	76COR	416	300	716	422	6	300	716
DDG 45	76COR	7284	600	7884	7410	126	600	7884
FF 1038	76COR	3923	20	3943	3923		20	3943
FF 1059	76COR	5975	400	6375	5977	2	400	6375
FFG 5	76COR	6704		6704	6704			6704
FFG 6	76COR	6956	150	7106	6983	27	150	7106
LPD 1	76COR	8081	40	8121	8121	40	40	8121
LSD 30	76COR	8716	130	8846	8748	32	130	8846

SHIP FUNDING ACTIVITY

PERIOD ENDING AUGUST 1975

ATLANTIC FLEET

FUNDING SCENARIO

REPORT FND S03001 DEC 16 1975

HULL	FND ACTVTY	PYRYR FND (1000)	CNTYR PLAN (1000)	CNTPLN ALLYRS (1000)	TOTFND ALLYRS TODTE	CNTYR FND TODTE	EST CNTYR RQMT	EST THOR OVHL
LSD 37	76COR	4722	260	4982	4882	160	260	4982
LST 1180	76COR	4491	90	4581	4561	70	90	4581
PG 93	76COR	625	27	652	632	7	27	652
SSN 585	76COR	28404	5442	33846	31704	3300	5442	33846
SSN 591	76COR	30497	3443	33940	31502	1005	3443	33940
SSN 604	76COR	22800	900	23700	23204	404	900	23700
SSN 664	76COR	16815	4	16819	16819	4	4	16819
SSN 670	76COR	21234	19	21253	21253	19	19	21253
SSN 674	76COR	20306	377	20683	20319	13	377	20683
SSN 676	76COR	16139	1500	17639	16145	6	1500	17639
SUBTOTAL								
		306507	15605	322112	312476	5969	15605	322112
DD 827	76CORR	4709	50	4759	4719	10	50	4759
DD 890	76CORR	4053	55	4108	4108	55	55	4108
MSO 429	76CORR	502	49	551	551	49	49	551
SUBTOTAL								
		9264	154	9418	9378	114	154	9418
CV 42	76SRA	437	5204	5641	3667	3230	5204	
CV 59	76SRA	800	5930	6730	2300	1500	5930	
CV 60	76SRA	1312	6977	8289	8289	6977	6977	
CV 62	76SRA	49	7905	7954	167	118	7905	
CV 67	76SRA	150	5984	6134	385	235	5984	
CVT 16	76SRA		2561	2561	200	200	2561	
SSBN 633	76SRA	919	2978	3897	3542	2623	2978	
SSBN 643	76SRA		4078	4078	500	500	4078	
SSN 682	76SRA		2889	2889	691	691	2889	
SSN 683	76SRA		2152	2152	223	223	2152	
SUBTOTAL								
		3667	46658	50325	19964	16297	46658	0

SHIP FUNDING ACTIVITY

PERIOD ENDING AUGUST 1975

ATLANTIC FLEET

FUNDING SCENARIO

REPORT FND S03001 DEC 16 1975

HULL	FND ACTVTY	PYRYR FND (1000)	CNTYR PLAN (1000)	CNTPLN ALLYRS (1000)	TOTFND ALLYRS TODTE	CNTYR FND TODTE	EST CNTYR RQMT.	EST THOR OVHL
		*****TOTAL****						
		361004	601995	909694	521846	160598	611068	859599

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS
QUERY S03
IF ASOFDATE IS 750831 AND PREFIX ENDATE IS 76?
REPORT FLT
HEAD 1 $SCHEDULE FOR RETURN TO THE FLEET$
HEAD 2 $FUNDING INFORMATION ONLY$
HEAD 3 $FUNDING SCENARIO$
$END

YOUR QUERY HAS BEEN ACCEPTED
%A


USER ABORT
QUERY RUN ABORTED IN ROUTINE SCRN

WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS
QUERY S03
IF ASOFDATE IS 750831 AND PREFIX ENDATE IS 76?
REPORT FLT
SORT ON ENDATE
HEAD 1 $SCHEDULE FOR RETURN TO THE FLEET$
HEAD 2 $FUNDING INFORMATION ONLY$
HEAD 3 $FUNDING SCENARIO$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY S03, QUESTION NO.    1, HAS        53 HITS-T
```

SHIP SCHEDULE REPORT

SCHEDULE FOR RETURN TO THE FLEET

FUNDING INFORMATION ONLY

FUNDING SCENARIO

REPORT FLT S03001 DEC 16 1975

END DATE	START DATE	HULL	SHIP	YARD	OVHL
JAN 76	AUG 75	LSD 29	PLYMOUTH ROCK	5	100
JAN 76	APR 75	FFG 5	PAGE	PH	100
JAN 76	JAN 75	SSN 676	BILLFISH	PT	100
FEB 76	OCT 75	CV 59	FORRESTAL	N	
FEB 76	AUG 74	SSN 591	SHARK	SP	100
FEB 76	JUL 75	LPH 9	GUAM	PH	100
FEB 76	NOV 75	PG 89	MARATHON	9	100
FEB 76	AUG 75	DD 821	JOHNSTON	3	100
MAR 76	FEB 76	CVT 16	LEXINGTON	SJ	
MAR 76	MAY 75	DDG 3	KING	N	100
MAR 76	NOV 75	ATF 162	SHAKORI	5	100
MAR 76	AUG 75	LST 1181	SUMTER	5	100
MAR 76	JUL 75	FF 1072	BLAKELY	C	100
APR 76	NOV 75	LPA 249	MARION	3	100
APR 76	AUG 75	LPD 14	TRENTON	5	100
APR 76	JAN 76	PG 86	ANTELOPE	NFLM	100
APR 76	JAN 76	PG 87	READY	NFLM	100
APR 76	SEP 75	DD 880	DYESS	3	100
APR 76	APR 75	CG 34	BIDDLE	BA	100
MAY 76	FEB 76	CV 67	KENNEDY	N	
MAY 76	APR 75	DDG 45	DEWEY	C	100
JUN 76	JAN 76	AD 19	YOSEMITE	SJ	100
JUN 76	AUG 75	DDG 23	BYRD	N	100
JUN 76	MAR 76	SSN 661	LAPON	NN	100
JUN 76	MAR 76	AGF 3	LA SALLE	SUBIC	100
JUN 76	APR 76	SSN 683	PARCHE	N	
JUL 76	JUL 74	SSN 585	SKIPJACK	SG	100
JUL 76	MAY 76	SSBN 643	BANCROFT	SG	
JUL 76	JUL 75	SSN 675	BLUEFISH	N	100
JUL 76	APR 76	MSO 442	FEARLESS	6	100
AUG 76	OCT 75	DDG 19	TATTNALL	C	100
AUG 76	MAY 76	MSO 441	EXULTANT	3.	100
AUG 76	JAN 76	AR 5	VULCAN	5	100
AUG 76	FEB 76	LSD 38	PENSACOLA	5	100
AUG 76	FEB 76	LST 1188	SAGINAW	5	100
AUG 76	OCT 75	FF 1075	TRIPPE	C	100
SEP 76	MAY 76	CV 62	INDEPENDENCE	N	
SEP 76	JAN 76	AS 34	CANOPUS	C	100
DATA BASE - BASE2					
SEP 76	JUN 76	MSO 448	ILLUSIVE		100
SEP 76	DEC 75	AE 28	SANTA BARBARA	SJ	100
SEP 76	APR 76	DD 866	CONE	SJ	100
SEP 76	OCT 75	DD 943	BLANDY	3	100
OCT 76	NOV 75	DDG 10	SAMPSON	PH	
OCT 76	APR 76	AS 36	SPEAR	N	100
OCT 76	MAR 76	LPD 12	SHREVEPORT	5	100
OCT 76	MAY 76	DD 822	MCCARD	SJ	100

SHIP SCHEDULE REPORT

SCHEDULE FOR RETURN TO THE FLEET

FUNDING INFORMATION ONLY

FUNDING SCENARIO

REPORT FLT S03001 DEC 16 1975

END DATE	START DATE	HULL	SHIP	YARD	OVHL
NOV 76	DEC 75	DDG 4	LAWRENCE	N	100
NOV 76	MAY 76	LPH 12	INCHON	'	100
NOV 76	DEC 75	FF 1068	VREELAND	PH	100
DEC 76	JUL 76	ARS 41	OPPORTUNE	5	
DEC 76	MAY 76	DD 842	FISKE	3	100
DEC 76	JAN 76	DD 933	BARRY	PH	100
DEC 76	JAN 76	DD 940	MANLEY	PH	100

CINCLANTFLT

Ship Funding Profile Scenario

Terminal Session and Report Samples

Attachment #5

```
NSRDC 6700 INTERCOM V4.2
DATE  12/17/75
TIME  13.13.02.

PLEASE LOGIN
LOGIN,PUWEWEAVER,1180600823.,SUP

COMMAND- ETL,500

COMMAND- COMRADE,SHARP



COMRADE        TIME:      13.13.47.
               DATE:      12/17/75

SHARP SUBSYSTEM ENTERED
???? REPDEF


 PLEASE INPUT REPORT DEFINITIONS.

ID BASE2
REPORT PRO
MHEAD $SHIP FUNDING PROFILE$
COL HEAD 1 = $ASOF/DATE$
COL HEAD 2 = $HULL$
COL HEAD 3 = $FND/ACTVTY$
COL HEAD 5 = $START/DATE$
COL HEAD 6 = $FND/DATE$
COL HEAD 7 = $FND/PRIOR/YEAR$
COL HEAD 8 = $FND/CNT/YEAR$
COL HEAD 9 = $EST/CNTYR/RCMT$
COL HEAD 10 = $EST/THOR/OVHL$
PRINT ASOFDATE, HULL, FNDACTVTY, STARTDATE, FNDATE, PYRYRFND,
CNTYRFND, ESTCNTYRCT, ESTOVHL
$END

REPORT DEFINITION ERROR SUMMARY

ID - BASE2



REPORT - PRO
NO ERRORS FOUND - REPORT DEFINITION ACCEPTED
???? REPDEF
```

```
PLEASE INPUT REPORT DEFINITIONS.

ID BASE2
REPORT YRD
MHEAD $YARD FUNDING INFORMATION$
COL HEAED 1 = $YARD$
COL HEAD 2 = $HULL$
COL HEAD 3 = $START/DATE$
COL HEAD 4 = $END/DATE$
COL HEAD 5 = $FND/ACTVTY$
COL HEAD 6 = $FND/PRIOR/YEAR$
COL HEAD 7 = $CNT/YEAR/FNDG$
COL HEAD 8 = $TOTAL/ALLYR/FNDG$
PRINT FCLTY, HULL, STARTDATE, ENDATE, FNDACTVTY, PYRYRFND,
CNTYRFND, TOTFND
$END

REPORT DEFINITION ERROR SUMMARY

ID - BASE2


REPORT - YRD
NO ERRORS FOUND - REPORT DEFINITION ACCEPTED
???? REPDEF
```

```
YOU HAVE SELECTED THE INTERACTIVE QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-NO


SELECT ONE OF THE FOLLOWING DATA BASES
18 LOGISTICS DOCUMENT DATA BANK
31 YURSO DATA BANK
32 MKRS-BASE-2
?? 32

PLEASE INPUT QUERY COMMANDS
QUERY ***
IF ASOFDATE IS 750131
AND ENDACTVTY IS 75CFY
AND PREFIX STARTDATE IS 7501?
PRINT HULL AND SHIP
HEAD 1 $JANUARY 1975 STARTS$
HEAD 2 $THE GREAT SHIP SCENARIO$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR P.

QUERY ***, QUESTION NO.    1, HAS           7 HITS-T
```

JANUARY I975 STARTS

THE GREAT SHIP SCENARIO

REPORT DET ***ØØ1 DEC 17 1975

HULL: DDG 2
SHIP: ADAMS

HULL: FFG 6
SHIP: FURER

HULL: SSN 676
SHIP: BILLFISH

HULL: LST 118Ø
SHIP: MANITOWOC

HULL: PG 98
SHIP: GRAND RAPIDS

HULL: PG 10Ø
SHIP: DOUGLAS

HULL: DD 937
SHIP: DAVIS

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS
QUERY *1*
IF HULL IS DDG 2 OR FFG 6 OR SSN 676 OR LST 1180 OR PG 98 OR
PG 100 OR DD 937
AND ASOFDATE IS BE TO 740731/750831
AND SUFFIX FNDACTVTY IS NE TO BDGT?
REPORT PRO
 SORT ON HULL AND ASOFDATE
HEAD 1 $JANUARY 1975 STARTS$
HEAD 2 $THE GREAT SHIP SCENARIO$
HEAD 3 $THIS IS HOW WE FUND THEM AND FIND THEM$
$END

YOUR QUERY HAS BEEN ACCEPTED

FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY *1*, QUESTION NO.    1, HAS         94 HITS-T
```

SHIP FUNDING PROFILE

JANUARY 1975 STARTS

THE GREAT SHIP SCENARIO

THIS IS HOW WE FUND THEM AND FIND THEM

REPORT PRO *1*001 DEC 17 1975

ASOF DATE	HULL	FND ACTVTY	START DATE	END DATE	FND PRIOR YEAR	FND CNT YEAR	EST CNTYR RQMT	EST THOR OVHL
JUL 31 74	DD 937	75CFY	JAN 75	SEP 75	150	1000	7930	8080
AUG 31 74	DD 937	75CFY	JAN 75	SEP 75	150	1000	8330	8480
SEP 30 74	DD 937	75CFY	JAN 75	SEP 75	150	1000	8330	8480
OCT 31 74	DD 937	75CFY	JAN 75	SEP 75	150	6197	8330	8480
NOV 30 74	DD 937	75CFY	JAN 75	SEP 75	150	6197	8330	8480
DEC 31 74	DD 937	75CFY	JAN 75	SEP 75	150	6443	8330	8480
JAN 31 75	DD 937	75CFY	JAN 75	SEP 75	150	6443	7830	7980
FEB 28 75	DD 937	75CFY	JAN 75	DEC 75	150	6662	7830	7980
MAR 31 75	DD 937	75CFY	JAN 75	DEC 75	150	6671	7830	7980
APR 30 75	DD 937	75CFY	JAN 75	DEC 75	150	7030	7830	7980
MAY 31 75	DD 937	75CFY	JAN 75	DEC 75	150	7083	7730	7880
JUN 30 75	DD 937	75CFY	JAN 75	DEC 75	150	7508	7508	7658
JUL 31 75	DD 937	76COR	JAN 75	DEC 75	7658		120	7778
AUG 31 75	DD 937	76COR	JAN 75	DEC 75	7658		120	7778
JUL 31 74	DDG 2	75CFY	JAN 75	NOV 75	405		7882	8287
AUG 31 74	DDG 2	75CFY	JAN 75	NOV 75	405		9482	9887
SEP 30 74	DDG 2	75CFY	JAN 75	NOV 75	405	20	9482	9887
OCT 31 74	DDG 2	75CFY	JAN 75	NOV 75	405	20	9082	9487
NOV 30 74	DDG 2	75CFY	JAN 75	NOV 75	405	4020	9082	9487
JAN 31 75	DDG 2	75CFY	JAN 75	NOV 75	405	5020	7882	8287
FEB 28 75	DDG 2	75CFY	FEB 75	DEC 75	405	5020	7882	8287
MAR 31 75	DDG 2	75CFY	FEB 75	DEC 75	405	5020	8060	8465
APR 30 75	DDG 2	75CFY	FEB 75	DEC 75	405	7080	8060	8465
MAY 31 75	DDG 2	75CFY	FEB 75	DEC 75	405	7227	8060	8465
JUN 30 75	DDG 2	75CFY	FEB 75	DEC 75	405	7584	7584	7989
JUL 31 75	DDG 2	76COR	FEB 75	DEC 75	7989		400	8389
AUG 31 75	DDG 2	76COR	FEB 75	DEC 75	7567		400	7967
JUL 31 74	FFG 6	75CFY	JAN 75	SEP 75	350	1000	7110	7460
AUG 31 74	FFG 6	75CFY	JAN 75	SEP 75	350	1000	7110	7460
SEP 30 74	FFG 6	75CFY	JAN 75	SEP 75	350	1018	7110	7460
NOV 30 74	FFG 6	75CFY	JAN 75	OCT 75	350	5579	7110	7460
DEC 31 74	FFG 6	75CFY	JAN 75	OCT 75	350	5579	7010	7360
JAN 31 75	FFG 6	75CFY	JAN 75	OCT 75	350	5579	7010	7360
FEB 28 75	FFG 6	75CFY	JAN 75	NOV 75	350	5875	7010	7360
MAR 31 75	FFG 6	75CFY	JAN 75	NOV 75	350	5980	7010	7360
APR 30 75	FFG 6	75CFY	JAN 75	NOV 75	350	6390	7010	7360
MAY 31 75	FFG 6	75CFY	JAN 75	NOV 75	350	6606	6910	7260
JUN 30 75	FFG 6	75CFY	JAN 75	NOV 75	350	6606	6606	6956
JUL 31 75	FFG 6	76COR	JAN 75	OCT 75	6956	11	11	6967
AUG 31 75	FFG 6	76COR	JAN 75	OCT 75	6956	27	150	7106
JUL 31 74	LST 1180	75CFY	JAN 75	MAY 75	560		4243	4803
AUG 31 74	LST 1180	75CFY	JAN 75	MAY 75	560	200	4243	4803
SEP 30 74	LST 1180	75CFY	JAN 75	JUN 75	560	200	4243	4803
OCT 31 74	LST 1180	75CFY	JAN 75	JUN 75	560	3680	4259	4819
NOV 30 74	LST 1180	75CFY	JAN 75	JUN 75	560	3680	4259	4819

SHIP FUNDING PROFILE

JANUARY 1975 STARTS

THE GREAT SHIP SCENARIO

THIS IS HOW WE FUND THEM AND FIND THEM

REPORT PRO *1*001 DEC 17 1975

ASOF DATE	HULL	FND ACTVTY	START DATE	END DATE	FND PRIOR YEAR	FND CNT YEAR	EST CNTYR RQMT	EST THOR OVHL
DEC 31 74	LST 1180	75CFY	JAN 75	JUN 75	560	3680	4519	5079
JAN 31 75	LST 1180	75CFY	JAN 75	JUN 75	560	3680	4084	4644
FEB 28 75	LST 1180	75CFY	JAN 75	JUN 75	560	3680	4084	4644
MAR 31 75	LST 1180	75CFY	JAN 75	JUN 75	560	3880	4484	5044
APR 30 75	LST 1180	75CFY	JAN 75	JUL 75	560	4390	4484	5044
MAY 31 75	LST 1180	75CFY	JAN 75	JUL 75	560	4390	4484	5044
JUN 30 75	LST 1180	75CFY	JAN 75	JUL 75	560	4405	4405	4965
JUL 31 75	LST 1180	76COR	JAN 75	AUG 75	4965	70	90	5055
AUG 31 75	LST 1180	76COR	JAN 75	AUG 75	4491	70	90	4581
JUL 31 74	PG 100	75CFY	JAN 75	MAR 75	10	25	526	536
AUG 31 74	PG 100	75CFY	JAN 75	MAR 75	10	50	526	536
SEP 30 74	PG 100	75CFY	JAN 75	MAR 75	10	50	526	536
OCT 31 74	PG 100	75CFY	JAN 75	MAR 75	10	350	526	536
NOV 30 74	PG 100	75CFY	JAN 75	MAR 75	10	350	526	536
DEC 31 74	PG 100	75CFY	JAN 75	MAR 75	10	450	526	536
JAN 31 75	PG 100	75CFY	JAN 75	MAR 75	10	475	521	531
FEB 28 75	PG 100	75CFY	JAN 75	MAR 75	10	475	505	515
MAR 31 75	PG 100	75CFY	JAN 75	MAR 75	10	485	505	515
APR 30 75	PG 100	75CFY	JAN 75	APR 75	10	485	505	515
MAY 31 75	PG 100	75CFY	JAN 75	APR 75	10	485	485	495
JUN 30 75	PG 100	75CFY	JAN 75	APR 74	10	487	487	497
AUG 31 75	PG 100	76AFR				10	25	
JUL 31 74	PG 98	75CFY	JAN 75	MAR 75	10	25	491	501
AUG 31 74	PG 98	75CFY	JAN 75	MAR 75	10	50	491	501
SEP 30 74	PG 98	75CFY	JAN 75	MAR 75	10	50	491	501
OCT 31 74	PG 98	75CFY	JAN 75	MAR 75	10	360	491	501

**, WHO IS RUNNING PWS JOBS.

**, WHO IS RUNNING PWS JOBS.

**, WHO IS RUNNING PWS JOBS.

**, WHO IS RUNNING PWS JOBS.

**, WHO IS RUNNING PWS JOBS.

**, WHO IS RUNNING PWS JOBS.

**, WHO IS RUNNING PWS JOBS.

**, WHO IS RUNNING PWS JOBS.

**, WHO IS RUNNING PWS JOBS.

NOV 30 74	PG 98	75CFY	JAN 75	MAR 75	10	360	491	501
DEC 31 74	PG 98	75CFY	JAN 75	MAR 75	10	440	491	501
JAN 31 75	PG 98	75CFY	JAN 75	MAR 75	10	504	521	531
FEB 28 75	PG 98	75CFY	JAN 75	MAR 75	10	505	505	515
MAR 31 75	PG 98	75CFY	JAN 75	MAR 75	10	495	505	515
APR 30 75	PG 98	75CFY	JAN 75	APR 75	10	485	495	515
MAY 31 75	PG 98	75CFY	JAN 75	APR 75	10	495	495	515
JUN 30 75	PG 98	75CFY	JAN 75	APR 75	10	495	495	505
AUG 31 75	PG 98	76AFR				10	25	
JUL 31 74	SSN 676	75CFY	JAN 75	JAN 76	1930		12654	15586
AUG 31 74	SSN 676	75CFY	JAN 75	JAN 76	1930		12654	15586

SHIP FUNDING PROFILE

JANUARY 1975 STARTS

THE GREAT SHIP SCENARIO

THIS IS HOW WE FUND THEM AND FIND THEM

REPORT PRO *1*001 DEC 17 1975

ASOF DATE	HULL	FND ACTVTY	START DATE	END DATE	FND PRIOR YEAR	FND CNT YEAR	EST CNTYR RQMT	EST THOR OVHL
MAY 31 75	SSN 676	75CFY	JAN 75	JAN 76	1930	10400	11900	15832
JUN 30 75	SSN 676	75CFY	JAN 75	JAN 76	1930	12207	12207	16139
JUL 31 75	SSN 676	76COR	JAN 75	JAN 76	16139	4	1500	17639
AUG 31 75	SSN 676	76COR	JAN 75	JAN 76	16139	6	1500	17639

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS
QUERY *2*
IF ASOFDATE IS 750630?
SORT ON FCLTY
REPORT YRD
HEAD 1 $INFORMATION IS ORDERED BY YARDS$
HEAD 2 $THE GREAT SHIP SCENARIO$
HEAD 3 $WHO GETS THE MONEY$
PRINT SUBTOTAL PYRYRFND, CNTYRFND, TOTFND ON FCLTY
PRINT TOTAL PYRYRFND CNTYRFND TOTFND
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY *2*, QUESTION NO.   1, HAS        180 HITS-T
```

YARD FUNDING INFORMATION

INFORMATION IS ORDERED BY YARD

THE GREAT SHIP SCENARIO

WHO GETS THE MONEY

REPORT YRD *2*001 DEC 17 1975

YARD	HULL	START DATE	END DATE	FND ACTVTY	FND PRIOR YEAR	CNT YEAR FNDG	TOTAL ALLYR FNDG
	DDG 19	OCT 75	AUG 76	75AFR	63	1278	1341
				SUBTOTAL			
					63	1278	1341
BA	CG 34	APR 75	APR 76	75CFY	530	7141	7671
				SUBTOTAL			
					530	7141	7671
C	DDG 18	JAN 74	DEC 74	75COR	5483	197	5680
C	DDG 11	JUN 74	MAY 75	75COR	6766	50	6816
C	FFG 6	JAN 75	NOV 75	75CFY	350	6606	6956
C	AS 34	JAN 76	SEP 76	75AFR		500	500
C	SSN 603	MAY 72	JAN 75	75COR	32463	735	33198
C	SSBN 609	NOV 72	MAR 75	75COR	33624	1697	35321
C	SSN 637	JUL 76	OCT 77	75AFR	600	1515	2115
C	SSN 653	AUG 75	NOV 76	75AFR	567	5020	5587
C	SSN 660	JAN 76	JAN 77	75AFR	96	2315	2411
C	SSN 669	APR 74	JUN 75	75COR	24622	226	24848
C	SSBN 631	DEC 76	APR 78	75AFR		408	408
C	SSN 681	OCT 75	NOV 75	75SRA		2103	2103
C	FF 1078	APR 76	FEB 77	75AFR		335	335
C	FF 1079	MAY 76	MAR 77	75AFR		283	283
C	DD 937	JAN 75	DEC 75	75CFY	150	7508	7658
C	DD 938	OCT 74	AUG 75	75CFY	349	7473	7822
C	DDG 45	APR 74	MAY 76	75CFY	205	7079	7284
C	FF 1047	SEP 74	JUL 75	75CFY	440	4858	5298
C	FF 1049	JUN 74	APR 75	75COR	4280	669	4949
C	FF 1072	JUL 75	MAR 76	75AFR	500	150	650
C	FF 1075	OCT 75	AUG 76	75AFR		516	516
				SUBTOTAL			
					110495	50243	160738

YARD FUNDING INFORMATION

INFORMATION IS ORDERED BY YARD

THE GREAT SHIP SCENARIO

WHO GETS THE MONEY

REPORT YRD *2*001 DEC 17 1975

YARD	HULL	START DATE	END DATE	FND ACTVTY	FND PRIOR YEAR	CNT YEAR FNDG	TOTAL ALLYR FNDG
LB	MSO 492	MAR 76	JUN 76	75AFPP	688	10	698
			SUBTOTAL				
					688	10	698
N	CVA 59	SEP 74	DEC 74	75SRA	383	5092	5475
N	CVA 59	JAN 77	DEC 77	75AFR		100	100
N	CVA 62	MAR 75	JUN 75	75SRA		4758	4758
N	CVA 66	DEC 74	OCT 75	75CFY	2641	28329	30970
N	CVA 67	FEB 76	MAY 76	75SPA		150	150
N	DDG 3	MAY 75	MAR 76	75CFY	401	8302	8703
N	DDG 4	DEC 75	NOV 76	75AFR		479	479
N	DDG 5	APR 74	DEC 74	75COR	6514	465	6979
N	DDG 17	OCT 74	AUG 75	75CFY	658	7410	8068
N	DDG 23	AUG 75	JUN 76	75AFR	5	1308	1313
N	SSN 668	SEP 73	JUL 74	75COR	18372	57	18429
N	SSN 670	JUL 74	MAY 75	75CFY	4585	16600	21234
N	SSN 675	JUL 75	JUL 76	75AFR	1713	1000	2713
N	AD 38	MAY 75	SEP 75	75CFY	15	907	922
N	AOE 3	JUL 74	FEB 75	75CFY	1000	7364	8364
N	LPH 12	MAY 75	NOV 76	75AFR	8	200	208
N	SSN 678	JAN 76	JAN 77	75AFR	15	914	929
N	SSN 680	MAR 75	APR 75	75SRA		1390	1390
N	FF 1081	JUL 76	MAY 77	75AFR		100	100
N	AOR 4	JUL 76	APR 77	75AFR		87	87
N	SSN 684	MAY 75	JUN 75	7.5SRA		1300	1300
N	DD 941	APR 74	DEC 75	75CFY		6938	6938
N	AS 36	APR 76	OCT 76	75AFR		125	125
			SUBTOTAL				
					36310	93375	129734
NN	SSN 661	JAN 76	APR 77	75AFR	41	1539	1580
NN	SSN 663	OCT 76	DEC 77	75AFR	44	841	885
			SUBTOTAL				
					85	2380	2465

YARD FUNDING INFORMATION

INFORMATION IS ORDERED BY YARD

THE GREAT SHIP SCENARIO

WHO GETS THE MONEY

REPORT YRD *2*001 DEC 17 1975

YARD	HULL	START DATE	END DATE	FND ACTVTY	FND PRIOR YEAR	CNT YEAR FNDG	TOTAL ALLYR FNDG
NONE	SSN 679	77		75AFR		837	837
			SUBTOTAL				
					0	837	837
PH	CVA 42	MAY 74	AUG 74	75SRA	5600	499	6099
PH	CG 4	NOV 76	JUL 77	75AFR		210	210
PH	DDG 2	FEB 75	DEC 75	75CFY	405	7584	7989
PH	DDG 6	APR 74	FEB 75	75COR	6450	260	6710
PH	DDG 10	NOV 75	OCT 76	75AFR		436	436
PH	FFG 5	APR 75	NOV 76	75CFY	50	6654	6704
PH	AOR 2	FEB 75	SEP 75	75CFY	74	6348	6422
PH	LPH 9	JUL 75	FEB 76	75AFR	30	2028	2058
PH	LPH 7	SEP 74	APR 75	75CFY	845	7868	8713
PH	LCC 20	JUL 74	FEB 75	75CFY	837	6247	7084

Report pages 4 through 7 removed for this appendix.

Final report page 8 follows.

YARD FUNDING INFORMATION

INFORMATION IS ORDERED BY YARD

THE GREAT SHIP SCENARIO

WHO GETS THE MONEY

REPORT YRD *2*001 DEC 17 1975

YARD	HULL	START DATE	END DATE	FND ACTVTY	FND PRIOR YEAR	CNT YEAR FNDG	TOTAL ALLYR FNDG
			SUBTOTAL				
					751	1839	2591
8	DD 763	AUG 74	APR 75	75CFYR	143	4348	4491
8	DD 827	FEB 75	JUL 75	75CFYR	107	4602	4709
			SUBTOTAL				
					250	8950	9200
9	PG 89	NOV 75	FEB 76	75AFPR			
			SUBTOTAL				
					0	0	0
			*****TOTAL****				
					364985	427135	794583

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS
QUERY *3*
IF FNDACTVTY IS 75BDGT OR
IF ASOFDATE IS 750630 AND FNDACTVTY IS 75CFY?
REPORT FST
SORT ON HULL AND ASOFDATE
HEAD 1 $THE APPORTIONMENT AND THE ACTUAL FUNDING AT YEAR END$
HEAD 2 $THE GREAT SHIP SCENARIO$
$END

YOUR QUERY HAS BEEN ACCEPTED

FOR EACH QUESTION, TYPE IN D, T, OR P.

QUERY    , QUESTION NO.   1, HAS         42 HITS-T
```

CURRENT YEAR FUNDING

THE APPORTIONMENT AND THE ACTUAL FUNDING AT YEAR END

THE GREAT SHIP SCENARIO

REPORT EST 001 DEC 17 1975

HULL	YARD	FND ACTVTY	START DATE	END DATE	CNT YR FNDING (1000)
AD 38	N	75CFY	MAY 75	SEP 75	907
AG 250	6	75CFY	JUN 75	OCT 75	630
AG 521	6	75CFY	JUN 75	OCT 75	550
AO 99	3	75CFY	MAR 75	SEP 75	4057
AOE 3	N	75CFY	JUL 74	FEB 75	7364
AOR 2	PH	75CFY	FEB 75	SEP 75	6348
APS 2	5	75CFY	MAY 75	NOV 75	3287
ARS 40	5	75CFY	SEP 74	MAR 75	1698
ARS 6	SJ	75CFY	JUN 75	NOV 75	572
CG 34	BA	75CFY	APR 75	APR 76	7141
CVA 66	N	75CFY	DEC 74	OCT 75	28329
DD 937	C	75CFY	JAN 75	DEC 75	7508
DD 938	C	75CFY	OCT 74	AUG 75	7473
DD 941	N	75CFY	APR 74	DEC 75	6938
DDG 17	N	75CFY	OCT 74	AUG 75	7410
DDG 2	PH	75CFY	FEB 75	DEC 75	7584
DDG 3	N	75CFY	MAY 75	MAR 76	8302
DDG 45	C	75CFY	APR 74	MAY 76	7079
FF 1038	PH	75CFY	MAR 75	SEP 75	3820
FF 1047	C	75CFY	SEP 74	JUL 75	4858
FF 1059	PH	75CFY	JUN 75	MAR 76	5825
FFG 5	PH	75CFY	APR 75	NOV 76	6654
FFG 6	C	75CFY	JAN 75	NOV 75	6606
LCC 20	PH	75CFY	JUL 74	FEB 75	6247
LKA 117	5	75CFY	JUL 74	FEB 75	6241
LPD 1	5	75CFY	AUG 74	AUG 75	7787
LPD 11	5	75CFY	DEC 74	MAY 75	5777
LPH 7	PH	75CFY	SEP 74	APR 75	7868
LSD 30	5	75CFY	FEB 75	SEP 75	8528
LSD 37	5	75CFY	JUN 75	DEC 75	4597
LST 1180	5	75CFY	JAN 75	JUL 75	4405
MSO 490	6	75CFY	OCT 74	DEC 74	267
PG 100	5	75CFY	JAN 75	APR 74	487
PG 92	5	75CFY	APR 74	JUL 75	440
PG 93	5	75CFY	JUN 75	OCT 75	625
PG 98	5	75CFY	JAN 75	APR 75	495
SSN 585	SG	75CFY	JUL 74	JUL 76	24690
SSN 591	SP	75CFY	AUG 74	FEB 76	23845
SSN 664	PT	75CFY	JUL 74	JUN 75	13596
SSN 670	N	75CFY	JUL 74	MAY 75	16600
SSN 674	PT	75CFY	OCT 74	OCT 75	17168
SSN 676	PT	75CFY	JAN 75	JAN 76	12207

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS
QUERY *3*
IF ASOFDATE IS 750630 OR ENDACTVTY IS 74BDGT OR ENDACTVTY IS
75BDGT?
REPORT QRY
SORT ON HULL AND ASOFDATE
HEAD 1 $APPORTIONMENT AND EST OVHL AT FY75 YEAR END$
HEAD 2 $THE GREAT SHIP SCENARIO$
HEAD 3 $NAVLIS RAISES QUESTIONS AND ANSWERS QUESTIONS$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY *3*, QUESTION NO.   1, HAS        312 HITS-T
```

BUDGET TO ACTUAL PERFORMANCE

APPORTIONMENT AND EST OVHL AT FY75 YEAR END

THE GREAT SHIP SCENARIO

NAVLIS RAISES QUESTIONS AND ANSWERS QUESTIONS

REPORT QRY *3*ØØ1 DEC 17 1975

HULL	END ACTVTY	BDGT LABOR (1ØØØ)	BDGT MATL (1ØØØ)	BDGT UNIT COST	EST THOP OVHL	OVHL
AD 18	74BDGT	175Ø	416	2166		
AD 19	75BDGT	199Ø	489	2479		
AD 38	75BDGT	2564	6Ø6	317Ø		
AD 38	75CFY				922	1ØØ
AE 21	74BDGT	1753	6Ø7	236Ø		
AE 21	75COR				3427	1ØØ
AE 27	74BDGT	1795	617	2412		
AE 27	75COR				26Ø9	1ØØ
AF 28	75AFR					
AF 58	74BDGT	2Ø96	4Ø2	2498		
AFS 2	75BDGT	2549	1Ø88	3637		
AFS 6	74BDGT	2716	6Ø7	3323		
AFS 6	75COR				3642	1ØØ
AG 25Ø	75CFY				63Ø	1ØØ
AG 521	75BDGT	179	1Ø2	281		
AG 521	75CFY				55Ø	1ØØ
AGF 3	75AFR					
AO 147	74BDGT	3483	967	445Ø		
AO 147	75COR				4485	1ØØ
AO 19	75AFR					
AO 98	74BDGT	4263	891	5154		
AO 98	75COR				4615	1ØØ
AO 99	75BDGT	5Ø19	1Ø79	6Ø98		
AO 99	75CFY				415Ø	1ØØ
AOE 3	75BDGT	6295	1296	7591		
AOE 3	75CFY				8364	1ØØ
AOE 4	75BDGT	551Ø	1134	6644		
AOE 4	75AFR					
AOG 5Ø	74BDGT	1462	4Ø1	1863		
AOR 2	75BDGT	54Ø9	11Ø1	651Ø		
AOR 2	75CFY				6422	1ØØ
AOR 4	75AFR					
APS 2	75CFY				3287	1ØØ
AR 5	75AFR					
ARS 4Ø	75BDGT	1Ø18	338	1356		
ARS 4Ø	75CFY				179Ø	1ØØ
ARS 6	75BDGT	958	332	129Ø		
ARS 6	75CFY				572	1ØØ
ARS 8	74BDGT	974	172	1146		
AS 34	75AFR					
AS 36	75AFR					
ASR 13	74BDGT	452	17	469		
ASR 13	75COR				1451	1ØØ
ASR 15	74BDGT	727	17	744		
ASR 15	75COR				3854	1ØØ

BUDGET TO ACTUAL PERFORMANCE

APPORTIONMENT AND EST OVHL AT FY75 YEAR END

THE GREAT SHIP SCENARIO

NAVLIS RAISES QUESTIONS AND ANSWERS QUESTIONS

REPORT QRY *3*001 DEC 17 1975

HULL	END ACTVTY	BDGT LABOR (1000)	BDGT MATL (1000)	BDGT UNIT COST	EST THOR OVHL	OVHL
ASR 16	74BDGT	727	17	744		
ATF 161	74BDGT	698	300	998		
ATF 161	75COR				1029	100
ATF 162	75AFR					
ATS 1	75AFR					
CG 10	74BDGT	8157	1275	9432		
CG 17	74BDGT	3048	414	3462		
CG 17	75COR				6149	100
CG 27	74BDGT	2875	375	3250		
CG 28	74BDGT	3533	592	4125		
CG 34	75BDGT	5597	895	6492		
CG 34	75CFY				7671	100
CG 4	75AFR					
CV 66	75BDGT	24730	5774	30504		
CVA 42	75SRA					
CVA 59	75SRA					
CVA 59	75AFR					
CVA 60	75SRA					
CVA 62	75SRA					
CVA 66	75CFY				32054	100
CVA 67	74BDGT	17165	3650	20815		
CVA 67	75SRA					
CVT 16	74BDGT	7959	2586	10545		
DD 714	75BDGT	4782	686	5468		
DD 714	75CFYR				4082	
DD 724	75BDGT	4500	638	5138		
DD 763	75BDGT	3749	731	4480		

Report abbreviated for this appendix.

CINCLANTFLT

NAVLIS Scenario

Terminal Session and Report Samples

Attachment #6

```
NSRDC 6700 INTERCOM V4.2
DATE  12/18/75
TIME  13.17.11.

PLEASE LOGIN
LOGIN,PUWEWEAVER,1180600823,SUP

COMMAND- ETL,500

COMMAND- COMRADE,SHARP



COMRADE         TIME:       13.18.08.
                DATE:       12/18/75

SHARP SUBSYSTEM ENTERED
????
REPDEF



 PLEASE INPUT REPORT DEFINITIONS.

ID BASE2
REPORT *I*
WIDTH = 130
MHEAD $A NAVLIS REPORT$

130
COL HEAD 1 = $HULL$
COL HEAD 2 = $SHIP$
COL HEAD 3 = $ASOF/DATE$
COL HEAD 4 = $START/DATE$
COL HEAD 5 = $END/DATE$
COL HEAD 6 = $TYPE/OVHL$
COL HEAD 7 = $ACT/MDAYS$
COL HEAD 8 = $FND/ACTVTY$
COL HEAD 9 = $BDGT/LBR/DAYS$
COL HEAD 10 = $BDGT/LBR/MNY$
COL HEAD 11 = $BDGT/MATL/MNY$
COL HEAD 12 = $BDGT/UNIT/COST$
COL HEAD 13 = $EST/THOR/OVHL$
COL HEAD 14 = $PCNT/OVHL$
COL HEAD 16 = $STD/OVHL/DUR$
COL HEAD 17 = $STD/OVHL/INTVL$
PRINT HULL, SHIP, ASOFDATE, STARTDATE, ENDATE, TYPEOVHL,
ACTLDAYS, FNDACTVTY, BDGTLDAYS, BDGTLMNY, BDGTMMNY,
BDGTUCOST, ESTOVHL, PERCNTOVHL, DURATN, INTRVL
$END
```

```
REPORT DEFINITION ERROR SUMMARY

ID - BASE2


EPORT - *I*
NO ERRORS FOUND - REPORT DEFINITION ACCEPTED
????
 EX ID=      CADC PFN=DATADEFFILE
 EX CY= 001  00001569 PRUS  $0003.92 /DAY
IQUERY


YOU HAVE SELECTED THE INTERACTIVE QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-NO


SELECT ONE OF THE FOLLOWING DATA BASES
18 LOGISTICS DOCUMENT DATA BANK
31 YURSO DATA BANK
32 MKRS-BASE-2
?? 32

PLEASE INPUT QUERY COMMANDS
```

```
QUERY *I*
IF HISTDATA IS OVHL OR
PREFIX FNDACTVTY IS BE TO 73/76 OR
SCDACTVTY IS PROJ?
REPORT *I*
SORT ON HULL AND ASOFDATE
HEAD 1 $DISPLAY OF DISTRIBUTED FILE INFORMATION$
HEAD 2 $SHIP - SCHEDULING
HEAD 2 $SHIP SCHEDULING - ATLANTIC FLEET$
HEAD 3 $BOW WAVE PROBLEM ANALYSIS$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY *I*, QUESTION NO.   1, HAS        3749 HITS-B
```

A NAVLIS REPORT

DISPLAY OF DISTRIBUTED FILE INFORMATION

SHIP SCHEDULING - ATLANTIC FLEET

BOW WAVE PROBLEM ANALYSIS

REPORT *I* *I*001 DEC 18 1975

HULL	SHIP	ASOF DATE	START DATE	END DATE	TYPE OVHL	ACT MDAYS	FND ACTVTY	BDGT LBR DAYS	BDGT LBR MNY	BDGT MATL MNY	BDGT UNIT COST	EST THOR OVHL	PCNT OVHL	STD OVHL DUR	STD OVHL INTVL
AD 17	PIEDMONT	APR 15 77												5.0	48.0
AD 18	SIERRA	APR 02 62	APR 02 62	AUG 15 62	F2	58338								2.0	33.0
AD 18	SIERRA	AUG 16 65	AUG 16 65	NOV 04 65	RO	13012								3.0	37.0
AD 18	SIERRA	SEP 15 69	SEP 15 69	DEC 15 69	RO	5877								3.0	48.0
AD 18	SIERRA	MAY 01 73	AUG 69	DEC 69			7480GT	14341	1750	416	2166				
AD 18	SIERRA	JAN 07 74	JAN 07 74	MAY 24 74	RO	20639								5.0	48.0
AD 18	SIERRA	APR 15 77	MAY 78											5.0	48.0
AD 19	YOSEMITE	APR 18 63	APR 18 63	JUN 18 63	RO	9495								2.0	33.0
AD 19	YOSEMITE	AUG 22 66	AUG 22 66	DEC 19 66	RO	19069								3.0	37.0
AD 19	YOSEMITE	JUL 16 70	JUL 16 70	OCT 16 70	RO	11744								3.0	48.0
AD 19	YOSEMITE	MAY 20 74	JUL 70	OCT 70			75BDGT		1990	489	2479				
AD 19	YOSEMITE	JUL 31 74	MAY 75	OCT 75			75CFY					3690			
AD 19	YOSEMITE	AUG 31 74	MAY 75	OCT 75			75CFY					3690			
AD 19	YOSEMITE	SEP 30 74	MAY 75	OCT 75			75CFY					3690	73		
AD 19	YOSEMITE	OCT 31 74					75AFR								
AD 19	YOSEMITE	NOV 30 74	JAN 76	JUN 76			75AFR						73		
AD 19	YOSEMITE	DEC 31 74	JAN 76	JUN 76			75AFR						73		
AD 19	YOSEMITE	JAN 31 75	JAN 76	JUN 76			75AFR						73		
AD 19	YOSEMITE	FEB 28 75	JAN 76	JUN 76			75AFR								
AD 19	YOSEMITE	MAR 31 75	JAN 76	JUN 76			75AFR								
AD 19	YOSEMITE	APR 30 75	JAN 76	JUN 76			75AFR								
AD 19	YOSEMITE	MAY 31 75	JAN 76	JUN 76			75AFR								
AD 19	YOSEMITE	JUN 01 75	JUL 70	OCT 70			76BDGT		3278	878	4156				
AD 19	YOSEMITE	JUL 31 75	JAN 76	JUN 76			76CFY					4156	100		
AD 19	YOSEMITE	AUG 31 75	JAN 76	JUN 76			76CFY					4063	100		
AD 38	PUGET SOUND	NOV 15 68	NOV 15 68	DEC 16 68	PS	1150								3.0	48.0
AD 38	PUGET SOUND	JUN 01 72					73BDGT		1615	325	1940				
AD 38	PUGET SOUND	MAY 20 74	FEB 71	MAY 71			75BDGT		2564	606	3170				
AD 38	PUGET SOUND	JUL 31 74	APR 75	OCT 75			75CFY					4350			
AD 38	PUGET SOUND	AUG 31 74	APR 75	OCT 75			75CFY					4350			
AD 38	PUGET SOUND	SEP 30 74	APR 75	DEC 75			75CFY					4350	77		
AD 38	PUGET SOUND	OCT 31 74	APR 75	DEC 75			75CFY					4350	77		
AD 38	PUGET SOUND	NOV 30 74	APR 75	OCT 75			75CFY					4350	77		
AD 38	PUGET SOUND	DEC 31 74	APR 75	OCT 75			75CFY					4350	77		
AD 38	PUGET SOUND	JAN 31 75	APR 75	OCT 75			75CFY					1050	77		
AD 38	PUGET SOUND	FEB 28 75	MAY 75	SEP 75			75CFY					1050	100		
AD 38	PUGET SOUND	MAR 31 75	MAY 75	SEP 75			75CFY					1050	100		
AD 38	PUGET SOUND	APR 30 75	MAY 75	SEP 75			75CFY					1050	100		
AD 38	PUGET SOUND	MAY 15 75	MAY 15 75	SEP 15 75	RO	39605								5.0	48.0
AD 38	PUGET SOUND	MAY 31 75	MAY 75	SEP 75			75CFY					1315	100		
AD 38	PUGET SOUND	JUN 30 75	MAY 75	SEP 75			75CFY					922	100		
AD 38	PUGET SOUND	JUL 31 75	MAY 75	SEP 75			76COR					922	100		
AD 38	PUGET SOUND	AUG 31 75	MAY 75	SEP 75			75COR					831	100		
AD 38	PUGET SOUND	APR 15 77	OCT 79											5.0	48.0
AE 21	SURIBACHI	MAY 26 64	MAY 26 64	DEC 15 65	C	125000								2.5	24.0

A NAVLIS REPORT

DISPLAY OF DISTRIBUTED FILE INFORMATION

SHIP SCHEDULING - ATLANTIC FLEET

BOW WAVE PROBLEM ANALYSIS

REPORT *I* *I*001 DEC 18 1975

HULL	SHIP	ASOF DATE	START DATE	END DATE	TYPE OVHL	ACT MDAYS	FND ACTVTY	BDGT LBR DAYS	BDGT LBR MNY	BDGT MATL MNY	BDGT UNIT COST	EST THCR OVHL	PCNT OVHL	STD OVHL DUR	STD OVHL INTVL
AE 21	SURIBACHI	DEC 15 65	DEC 15 65	JAN 28 66	FO	1772								3.0	37.0
AE 21	SURIBACHI	APR 15 66	APR 15 66	JUN 06 66	PS	9925								3.0	37.0
AE 21	SURIBACHI	JAN 16 69	JAN 16 69	APR 29 69	RO	12247								3.0	37.0
AE 21	SURIBACHI	MAY 01 73	JAN 69	APR 69			74BDGT		1753	607	2360				
AE 21	SURIBACHI	JUN 24 74	JUN 24 74	FEB 08 75	RO	38910								5.0	48.0
AE 21	SURIBACHI	AUG 31 74	JUN 74	JAN 75			75COR					3467			
AE 21	SURIBACHI	SEP 30 74	JUN 74	JAN 75			75COR					3532	81		
AE 21	SURIBACHI	OCT 31 74	JUN 74	JAN 75			75COR					3532	81		
AE 21	SURIBACHI	NOV 30 74	JUN 74	FEB 75			75COR					3318	81		
AE 21	SURIBACHI	DEC 31 74	JUN 74	FEB 75			75COR					3338	81		
AE 21	SURIBACHI	JAN 31 75	JUN 74	FEB 75			75COR					3437	81		
AE 21	SURIBACHI	FEB 28 75	JUN 74	FEB 75			75COR					3437	100		
AE 21	SURIBACHI	MAR 31 75	JUN 74	FEB 75			75COR					3437	100		
AE 21	SURIBACHI	APR 30 75	JUN 74	FEB 75			75COR					3437	100		
AE 21	SURIBACHI	MAY 31 75	JUN 74	FEB 75			75COR					3437	100		
AE 21	SURIBACHI	JUN 30 75	JUN 74	FEB 75			75COR					3427	100		
AE 21	SURIBACHI	APR 15 77	JAN 79											5.0	48.0
AE 23	NITRO	JUL 31 75	OCT 76	MAY 77			76AFR								
AE 23	NITRO	AUG 31 75	OCT 76	MAY 77			76AFR								
AE 23	NITRO	APR 15 77	JAN 76											5.0	48.0
AE 27	BUTTE	FEB 21 66	FEB 21 66	NOV 29 68	NC	310000								3.0	37.0
AE 27	BUTTE	FEB 10 69	FEB 10 69	NOV 14 69	FO	31700								3.0	37.0
AE 27	BUTTE	FEB 13 70	FEB 13 70	APR 15 70	PS	3123								3.0	37.0
AE 27	BUTTE	MAY 01 73	JAN 69	APR 69			74BDGT		1795	617	2412				
AE 27	BUTTE	FEB 11 74	FEB 11 74	AUG 12 74	RO	33345								5.0	48.0
AE 27	BUTTE	JUL 31 74	FEB 74	AUG 74			75COR					2559			
AE 27	BUTTE	AUG 31 74	FEB 74	AUG 74			75COR					2559			
AE 27	BUTTE	SEP 30 74	FEB 74	AUG 74			75COR					2615			
AE 27	BUTTE	OCT 31 74	FEB 74	AUG 74			75COR					2615			
AE 27	BUTTE	NOV 30 74	FEB 74	AUG 74			75COR					2615			
AE 27	BUTTE	DEC 31 74	FEB 74	AUG 74			75COR					2615			
AE 27	BUTTE	JAN 31 75	FEB 74	AUG 74			75COR					2615			
AE 27	BUTTE	FEB 28 75	FEB 74	AUG 74			75COR					2615	100		
AE 27	BUTTE	MAR 31 75	FEB 74	AUG 74			75COR					2615	100		
AE 27	BUTTE	APR 30 75	FEB 74	AUG 74			75COR					2615	100		
AE 27	BUTTE	MAY 31 75	FEB 74	AUG 74			75COR					2615	100		
AE 27	BUTTE	JUN 30 75	FEB 74	AUG 74			75COR					2609	100		
AE 27	BUTTE	APR 15 77	AUG 78											5.0	48.0
AE 28	SANTA BARBARA	AUG 08 66	AUG 08 66	JUL 02 70	NC	310000								3.0	37.0
AE 28	SANTA BARBARA	JUL 01 70	JUL 01 70	SEP 04 70	FO	3397								3.0	37.0
AE 28	SANTA BARBARA	APR 21 71	APR 21 71	JUN 25 71	PS	4480								3.0	37.0
AE 28	SANTA BARBARA	AUG 31 74	JAN 76	OCT 76			75AFR								
AE 28	SANTA BARBARA	SEP 30 74		JAN 76			75AFR								
AE 28	SANTA BARBARA	OCT 31 74	JAN 76	OCT 76			75AFR								
AE 28	SANTA BARBARA	NOV 30 74	DEC 75	SEP 76			75AFR								

A NAVLIS REPORT

DISPLAY OF DISTRIBUTED FILE INFORMATION

SHIP SCHEDULING - ATLANTIC FLEET

BOW WAVE PROBLEM ANALYSIS

REPORT *I* *I*001 DEC 18 1975

HULL	SHIP	ASOF DATE	START DATE	END DATE	TYPE OVHL	ACT MDAYS	FND ACTVTY	BDGT LBR DAYS	BDGT LBR MNY	BDGT MATL MNY	BDGT UNIT COST	EST THOR OVHL	PCNT OVHL	STD OVHL DUR	STD OVHL INTVL
AE 28	SANTA BARBARA	DEC 31 74	DEC 75	SEP 76			75AFR								
AE 28	SANTA BARBARA	JAN 31 75	DEC 75	SEP 76			75AFR								
AE 28	SANTA BARBARA	FEB 28 75	DEC 75	SEP 76			75AFR								
AE 28	SANTA BARBARA	MAR 31 75	DEC 75	SEP 76			75AFR								
AE 28	SANTA BARBARA	APR 30 75	DEC 75	SEP 76			75AFR								
AE 28	SANTA BARBARA	MAY 31 75	DEC 75	SEP 76			75AFR								
AE 28	SANTA BARBARA	JUN 01 75	FEB 71				76BDGT		4038	1396	5434				
AE 28	SANTA BARBARA	JUN 30 75	DEC 75	SEP 76			75AFR								
AE 28	SANTA BARBARA	JUL 31 75	DEC 75	SEP 76			76CFY					5359	100		
AE 28	SANTA BARBARA	AUG 31 75	DEC 75	SEP 76			76CFY					5205	100		
AE 28	SANTA BARBARA	DEC 03 75	DEC 09 75	SEP 08 76	RO	58135								5.0	48.0
AE 34	MOUNT BAKER	DEC 01 69	DEC 01 69	JUL 06 72	NC	192000								3.0	37.0
AE 34	MOUNT BAKER	JUL 10 72	JUL 10 72	SEP 19 72	FO	7335								4.0	48.0
AE 34	MOUNT BAKER	MAY 01 73	MAY 01 73	AUG 06 73	PS	12433								4.0	48.0
AE 34	MOUNT BAKER	APR 15 77	JUL 77											5.0	48.0
AF 21	SURIBACHI	JUL 31 74	JUN 74	JAN 75			75COR					3467			
AF 58	RIGEL	OCT 15 63	OCT 15 63	JAN 06 64	RO	10099								2.5	24.0
AF 58	RIGEL	JAN 30 67	JAN 30 67	MAY 08 67	RO	13986								3.0	37.0
AF 58	RIGEL	SEP 16 70	SEP 16 70	DEC 17 70	RO	20439								3.0	37.0
AF 58	RIGEL	MAY 01 73	SEP 70	DEC 70			74BDGT	18883	2096	402	2498				
AF 58	RIGEL	OCT 05 73	OCT 05 73	FEB 05 74	RO	31300								3.0	37.0
AF 58	RIGEL	APR 15 77	MAR 77											3.0	37.0
AF 58	RIGEL	APR 15 77	MAR 80											3.0	37.0
AFS 2	SYLVANIA	OCT 15 64	OCT 15 64	FEB 09 65	PS	21809								3.0	37.0
AFS 2	SYLVANIA	MAR 02 70	MAR 02 70	JUL 13 70	RO	33429								3.0	37.0
AFS 2	SYLVANIA	MAY 20 74	MAR 70	JUL 70			75BDGT		2549	1088	3637				
AFS 2	SYLVANIA	JUL 31 74	MAY 75	OCT 75			75CFY					3854			
AFS 2	SYLVANIA	AUG 31 74	MAY 75	OCT 75			75CFY					3854			
AFS 2	SYLVANIA	SEP 30 74	MAY 75	OCT 75			75CFY					3714			
AFS 2	SYLVANIA	OCT 31 74	MAY 75	OCT 75			75CFY					3714			

Report abbreviated for this appendix.

CINCLANTFLT

Ship Identification Scenario

Terminal Session and Report Samples

Attachment #7

```
NSRDC 6700 INTERCOM V4.2
DATE  12/23/75
TIME  15.59.12.

PLEASE LOGIN
LOGIN,PUWEWEAVER,1180600823,SUP

COMMAND- ETL,500

COMMAND- COMRADE,SHARP


COMRADE         TIME:      15.59.49.
                DATE:      12/23/75

SHARP SUBSYSTEM ENTERED
???? IQUERY


YOU HAVE SELECTED THE INTERACTIVE QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-NO


SELECT ONE OF THE FOLLOWING DATA BASES
18 LOGISTICS DOCUMENT DATA BANK
31 YURSO DATA BANK
32 MKRS-BASE-2
?? 32

PLEASE INPUT QUERY COMMANDS

QUERY DO
IF SHIPID IS OILER?
IF SHIPID IS CARRIER?
IF SHIP ID IS GARBAGE?
IF SHIPID IS MINESWEEPER?
IF SHIPID IS GASOLINE?
IF SHIPID IS SSN?
IF SHIP ID IS PATROL?
IF SHIPID IS TENDER?


IF PARTIAL SHIPS IS (9)FOX?


IF PREFIX SHIPS IS SSN 673?
$END

YOUR QUERY HAS BEEN ACCEPTED
```

```
FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY DO , QUESTION NO.   1, HAS           2 HITS-T

QUERY DO , QUESTION NO.   2, HAS           6 HITS-T

QUERY DO , QUESTION NO.   3, HAS           2 HITS-T

QUERY DO , QUESTION NO.   4, HAS          15 HITS-T

QUERY DO , QUESTION NO.   5, HAS           3 HITS-T

QUERY DO , QUESTION NO.   6, HAS           2 HITS-T

QUERY DO , QUESTION NO.   7, HAS          13 HITS-T

QUERY DO , QUESTION NO.   8, HAS           9 HITS-T

QUERY DO , QUESTION NO.   9, HAS           1 HITS-T

QUERY DO , QUESTION NO.  10, HAS           1 HITS-T
```

These reports removed for this appendix.

```
PROC/DATE: 751Ø15
RCDID: 2ØØØØ3Ø34
SHIPID: AO
        OILER
SHIPS: AO 144    MISSISSINEWA     Ø59Ø4
       AO 147    TRUCKEE          Ø59Ø7
       AO 98     CALOOSAHATCHEE   Ø4848

PROC/DATE: 751Ø15
RCDID: 2ØØØØ3Ø37
SHIPID: AOR
        REPLENISHMENT
        OILER
SHIPS: AOR 2     MILWAUKEE        Ø585Ø
       AOR 4     SAVANNAH         2Ø123
       AOR 6     KALAMAZOO        2Ø125
```

```
PROC/DATE: 751015
RCDID: 200003060
SHIPID: ATC
        ARMORED
        TROOP
        CARRIER
SHIPS: NONE

PROC/DATE: 751015
RCDID: 200003082
SHIPID: CVA
        ATTACK
        AIRCRAFT
        CARRIER
SHIPS: CVA 42   ROOSEVELT        03342
       CVA 59   FORRESTAL        03359
       CVA 60   SARATOGA         03360
       CVA 62   INDEPENDENCE     03362
       CVA 66   AMERICA          03366
       CVA 67   KENNEDY          03367

PROC/DATE: 751015
RCDID: 200003083
SHIPID: CVAN
        ATTACK
        AIRCRAFT
        CARRIER
        NUCLEAR
SHIPS: CVAN     NIMITZ           03368

PROC/DATE: 751015
RCDID: 200003084
SHIPID: CVE
        AIRCRAFT
        CARRIER
        ESCORT
SHIPS: NONE

PROC/DATE: 751015
RCDID: 200003085
SHIPID: CVS
        AIRCRAFT
        CARRIER
        ASW
SHIPS: CVS 16   LEXINGTON        03318
```

REPORT DFT DO 002

DEC 23 1975

PROC/DATE: 751015
RCDID: 200003086
SHIPID: CVT
AIRCRAFT
CARRIER
TRAINING
SHIPS: NONE

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS
QUERY 1.1
IF PARTIAL SHIPS IS (9)SUNBIRD?
PRINT ALL
HEAD 1 $THE SHIP IS THE *BIRD$
HEAD 2 $SHARP IS TO THE *$
HEAD 3 $THE POINT IS SHARP$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY 1   , QUESTION NO.   1, HAS          1 HITS-T
```

THE SHIP IS THE *BIRD

SHARP IS TO THE *

THE POINT IS SHARP

REPORT DFT 1 001 DEC 23 1975

PROC/DATE: 751015
RCDID: 200003058
SHIPID: ASR
SUBMARINE
RESCUE
SHIP

SHIPS:	ASR 13	KITTIWAKE	04712
	ASR 14	PETREL	04713
	ASR 15	SUNBIRD	04714
	ASR 16	TRINGA	04715
	ASR 22	ORTOLAN	20144

CINCLANTFLT

Scenario II

Terminal Sessions and Sample Reports

Attachment #8

```
19/7=.75
TIME  14.41.01.

PLEASE LOGIN
LOGIN,PUWEWEAVER,1180600823,SUP

WAITING FOR INPUT
IQUERY


PLEASE ANSWER YES OR NO- NO


???? IQUERY


YOU HAVE SELECTED THE INTERACTIVE QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-NO


SELECT ONE OF THE FOLLOWING DATA BASES
18 LOGISTICS DOCUMENT DATA BANK
31 YURSO DATA BANK
32 MKRS-BASE-2
?? 32


PLEASE INPUT QUERY COMMANDS

QUERY WES
IF ASOFDATE IS 750630 AND PREFIX ENDATE IS LE TO 7506 AND PREFIX
HULL IS DDG?
IF ASOFDATE IS 750831 AND PREFIX ENDATE IS LE TO 750831 AND PREFIX
HUL;L
HULL IS DDG?
$END

YOUR QUERY HAS BEEN ACCEPTED



FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY WES, QUESTION NO.   1, HAS          5 HITS-T

QUERY WES, QUESTION NO.   2, HAS          1 HITS-T
```

REPORT DFT WESØØ1 DEC 29 1975

ASOFDATE: 75Ø63Ø
CDR: SHARP
CNTPLNTOT: 568Ø
CNTYRFND: 197
CNTYRPLN: 197
ENDATE: 7412
ESTCNTYRQ: 197
ESTOVHL: 568Ø
FCLTY: C
FNDACTVTY: 75COR
HULL: DDG 18
NAVY: CINCLANTFLT
PERCNTOVH: 1ØØ
PROC/DATE: 751Ø29
PYRYRFND: 5483
RCDID: Ø46485227
SHIP: SEMMES
STARTDATE: 74Ø1
TOTFND: 568Ø
TYCOM: 2
UIC: 4648

ASOFDATE: 75Ø63Ø
CDR: SHARP
CNTPLNTOT: 6979
CNTYRFND: 465
CNTYRPLN: 465
ENDATE: 7412
ESTCNTYRQ: 465
ESTOVHL: 6979
FCLTY: N
FNDACTVTY: 75COR
HULL: DDG 5
NAVY: CINCLANTFLT
PERCNTOVH: 1ØØ
PROC/DATE: 751Ø29
PYRYRFND: 6514
RCDID: Ø46715224
SHIP: RICKETTS
STARTDATE: 74Ø4
TOTFND: 6979
TYCOM: 2
UIC: 4671

REPORT DFT WESØØ1 DEC 29 1975

ASOFDATE: 75Ø63Ø
CDR: SHARP
CNTPLNTOT: 671Ø
CNTYRFND: 26Ø
CNTYRPLN: 26Ø
ENDATE: 75Ø2
ESTCNTYRQ: 26Ø
ESTOVHL: 671Ø
FCLTY: PH
FNDACTVTY: 75COR
HULL: DDG 6
NAVY: CINCLANTFLT
PERCNTOVH: 1ØØ
PROC/DATE: 751Ø29
PYRYRFND: 645Ø
RCDID: Ø46725225
SHIP: BARNEY
STARTDATE: 74Ø4
TOTFND: 671Ø
TYCOM: 2
UIC: 4672

ASOFDATE: 75Ø63Ø
CDR: SHARP
CNTPLNTOT: 6816
CNTYRFND: 5Ø
CNTYRPLN: 5Ø
ENDATE: 75Ø5
ESTCNTYRQ: 5Ø
ESTOVHL: 6816
FCLTY: C
FNDACTVTY: 75COR
HULL: DDG 11
NAVY: CINCLANTFLT
PERCNTOVH: 1ØØ
PROC/DATE: 751Ø29
PYRYRFND: 6766
RCDID: Ø46775226
SHIP: SELLERS
STARTDATE: 74Ø6
TOTFND: 6816
TYCOM: 2
UIC: 4677

REPORT DFT WESØØ1 DEC 29 1975

ASOFDATE: 75Ø63Ø
CDR: SHARP
CNTPLNTOT: 93Ø7
CNTYRFND: 325
CNTYRPLN: 325
ENDATE: 75Ø1
ESTCNTYRQ: 325
ESTOVHL: 93Ø7
FCLTY: PH
FNDACTVTY: 75COR
HULL: DDG 37
NAVY: CINCLANTFLT
PERCNTOVH: 1ØØ
PROC/DATE: 751Ø29
PYRYRFND: 8982
RCDID: 522315222
SHIP: FARRAGUT
STARTDATE: 7311
TOTFND: 93Ø7
TYCOM: 2
UIC: 52231

REPORT DFT WESØØ2 DEC 29 1975

ASOFDATE: 75Ø831
CDR: SHARP
CNTPLNTOT: 8Ø88
CNTYRFND: 19
CNTYRPLN: 2Ø
ENDATE: 75Ø8
ESTCNTYRQ: 2Ø
ESTOVHL: 8Ø88
FCLTY: N
FNDACTVTY: 76COR
HULL: DDG 17
NAVY: CINCLANTFLT
PERCNTOVH: 1ØØ
PROC/DATE: 7512Ø9
PYRYRFND: 8Ø68
RCDID: Ø46835587
SHIP: CONYNGHAM
STARTDATE: 751Ø
TOTFND: 8Ø37
TYCOM: 2
UIC: 4683

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- NO


???? REPDEF


 PLEASE INPUT REPORT DEFINITIONS.

ID BASE2
REPORT WES
WIDTH = 130
MHEAD $MONTHLY FUNDING PROFILES$

130
COL HEAD 1 = $HULL$
COL HEAD 2 = $ASOF/DATE$
COL HEAD 3 = $START/DATE$
COL HEAD 4 = $END/DATE$
COL HEAD 5 = $FCLTY$
COL HEAD 6 = $PYRYR/FND$
COL HEAD 7 = $CHORD$
COL HEAD 8 = $CNTYR/PLAN$
COL HEAD 9 = $CNT/PLAN/TOTAL$
COL HEAD 10 = $TOTFND/ALLYR/TODTE$
COL HEAD 11 = $CNT/YEAR/FNDG$
COL HEAD 12 = $EST/CNTYR/RQMT$
COL HEAD 1Q
COL HEAD 13 = $EST/OVHL$
COL HEAD 14 = $PERCNT/OVHL$
PRINT HULL, ASOFDATE, STARTDATE, ENDATE, FCLTY, PYRYRFND, CHORD,
CNTYRPLN, CNTPLNTOT, TOTFND, CNTYRFND, ESTCNTYRQT, ESTOVHL,
PERCNTOVHL
$END

REPORT DEFINITION ERROR SUMMARY

ID - BASE2



REPORT - WES
NO ERRORS FOUND - REPORT DEFINITION ACCEPTED
????
QUERY

 EX ID=      CADC PFN=DATADEFFILE
 EX CY= 001  00001617 PRUS  $0004.04 /DAY


YOU HAVE SELECTED THE SHARP QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-NO


DO YOU WISH TO SUBMIT A QUERY RUN?-NO

???? IQUERY
```

```
YOU HAVE SELECTED THE INTERACTIVE QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-NO


SELECT ONE OF THE FOLLOWING DATA BASES
18 LOGISTICS DOCUMENT DATA BANK
31 YURSO DATA BANK
32 MKRS-BASE-2
?? 32


PLEASE INPUT QUERY COMMANDS

QUERY WES
IF SUFFIX FNDACTVTY IS NE TO BDGT AND
ASOFDATE IS BE TO 740731/750831 AND
HISTDATE IS NE TO OVHL AND
SCDACTVTY IS NE TO PROJ AND

FATAL ERROR--SEARCH VALUE TOO LARGE        --QUERY WES, QUESTION 001
ABOVE QUESTION WAS NOT PROCESSED. PLEASE EXAMINE FOR SYNTAX ERROR
AND CONTENT, RE-ENTER IF POSSIBLE

IF SUFFIX FNDACTVTY IS NE TO BDGT AND
ASOFDATE IS BE TO 740731/750831 AND
HISTDATA IS NE TO OVHL AND
SCDACTVTY IS NE TO PROJ AND
PREFIX HULL IS DDG AND
SUFFIX HULL IS 18 OR 5 OR 6 OR 11 OR 37 OR17
SUFFIX HULL IS 18 OR 5 OR 6 OR 11 OR 37 OR 17?
REPORT WES
SORT ON HULL AND ASOFDATE
HEAD 1 $CLASS DDG SHIPS - ATLANTIC FLEET$
$END

YOUR QUERY HAS BEEN ACCEPTED
```

```
NO HITS
ALTHOUGH INDIVIDUAL SEARCH CRITERIA WERE MET, NO RECORD ON THE
DATA BASE SATISFIED THE USER'S COMBINATION OF REQUIREMENTS.

WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS
QUERY WES
IF SUFFIX FNDACTVTY IS COR AND PREFIX HULL IS DDG AND SUFFIX HULL IS
18 OR 5 OR 6 OR 11 OR 37 OR 17?
REPORT WES
SORT ON HULL AND ASOFDATE
HEAD 1 $CLASS DDG SHIPS - ATLANTIC FLEET$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY WES, QUESTION NO.   1, HAS        55 HITS-B
```

MONTHLY FUNDING PROFILE

CLASS DDG SHIPS - ATLANTIC FLEET

REPORT WES WES001 DEC 29 1975

HULL	ASOF DATE	START DATE	END DATE	FCLTY	PYRYR FND	CHORD	CNTYR PLAN	CNT PLAN TOTAL	TOTFND ALLYR TODTE	CNT YEAR FNDG	EST CNTYR RQMT	EST OVHL	PERCNT OVHL
DDG 11	SEP 30 74	JUN 74	JUL 75	C	6534		8	6542	6534		108	6642	98
DDG 11	OCT 31 74	JUN 74	JUL 75	C	6534		38	6572	6542	8	38	6572	98
DDG 11	NOV 30 74	JUN 74	JUL 75	C	6534		38	6572	6542	8	38	6572	98
DDG 11	DEC 31 74	JUN 74	JUL 75	C	6534		8	6542	6542	8	8	6542	98
DDG 11	JAN 31 75	JUN 74	JUL 75	C	6534		8	6542	6542	8	258	6792	98
DDG 11	FEB 28 75	JUN 74	JUL 75	C	6534		8	6542	6542	8	258	6792	96
DDG 11	MAR 31 75	JUN 74	JUL 75	C	6534		280	6814	6542	8	280	7046	97
DDG 11	APR 30 75	JUN 74	JUL 75	C	6766		280	7046	6801	35	280	7046	100
DDG 11	MAY 31 75	JUN 74	JUL 75	C	6766		50	6816	6816	50	50	6816	100
DDG 11	JUN 30 75	JUN 74	MAY 75	C	6766		50	6816	6816	50	50	6816	100
DDG 17	JUL 31 75	OCT 75	AUG 75		8068		20	8088	8087	19	20	8088	100
DDG 17	AUG 31 75	OCT 75	AUG 75	N	8068		20	8088	8087	19	20	8088	100
DDG 18	SEP 30 74	JAN 74	DEC 74	C	5483		219	5702	5702	219	319	5802	98
DDG 18	OCT 31 74	JAN 74	DEC 74	C	5483		254	5737	5702	219	254	5737	98
DDG 18	NOV 30 74	JAN 74	DEC 74	C	5483		254	5737	5702	219	254	5737	98
DDG 18	DEC 31 74	JAN 74	DEC 74	C	5483		254	5737	5702	219	254	5737	98
DDG 18	JAN 31 75	JAN 74	DEC 74	C	5483		197	5680	5680	197	197	5680	98
DDG 18	FEB 28 75	JAN 74	DEC 74	C	5483		197	5680	5680	197	197	5680	100
DDG 18	MAR 31 75	JAN 74	DEC 74	C	5483		197	5680	5680	197	197	5680	100
DDG 18	APR 30 75	JAN 74	DEC 74	C	5483		197	5680	5680	197	197	5680	100
DDG 18	MAY 31 75	JAN 74	DEC 74	C	5483		197	5680	5680	197	197	5680	100
DDG 18	JUN 30 75	JAN 74	DEC 74	C	5483		197	5680	5680	197	197	5680	100
DDG 37	JUL 31 74	NOV 73	DEC 74	PH	8513			8513	8513		500	9013	
DDG 37	AUG 31 74	NOV 73	DEC 74	PH	8513			8513	8513		775	9288	
DDG 37	SEP 30 74	NOV 73	DEC 74	PH	8513		200	8713	8513	0	350	8863	98
DDG 37	OCT 31 74	NOV 73	DEC 74	PH	8982		155	9137	9117	135	155	9137	
DDG 37	NOV 30 74	NOV 73	JAN 75	PH	8982		155	9137	9137	155	155	9137	98
DDG 37	DEC 31 74	NOV 73	JAN 75	PH	8982		155	9137	9137	155	155	9137	98
DDG 37	JAN 31 75	NOV 73	JAN 75	PH	8982		325	9307	9307	325	325	9307	98
DDG 37	FEB 28 75	NOV 73	JAN 75	PH	8982		325	9307	9307	325	325	9307	100
DDG 37	MAR 31 75	NOV 73	JAN 75	PH	8982		325	9307	9307	325	325	9307	100
DDG 37	APR 30 75	NOV 73	JAN 75	PH	8982		325	9307	9307	325	325	9307	100
DDG 37	MAY 31 75	NOV 73	JAN 75	PH	8982		325	9307	9307	325	325	9307	100
DDG 37	JUN 30 75	NOV 73	JAN 75	PH	8982		325	9307	9307	325	325	9307	100
DDG 45	JUL 31 75	APR 75	MAY 76	C	7284		600	7884	7410	126	600	7884	100
DDG 45	AUG 31 75	APR 75	MAY 76	C	7284		600	7884	7410	126	600	7884	100
DDG 5	SEP 30 74	APR 74	DEC 74	N	6514		400	6914	6514		400	6914	
DDG 5	OCT 31 74	APR 74	DEC 74	N	6514		434	6948	6514	0	434	6948	
DDG 5	NOV 30 74	APR 74	DEC 74	N	6514		434	6948	6948	434	434	6948	
DDG 5	DEC 31 74	APR 74	DEC 74	N	6514		465	6979	6979	465	465	6979	
DDG 5	JAN 31 75	APR 74	DEC 74	N	6514		465	6979	6979	465	465	6979	
DDG 5	FEB 28 75	APR 74	DEC 74	N	6514		465	6979	6979	465	465	6979	100
DDG 5	MAR 31 75	APR 74	DEC 74	N	6514		465	6979	6979	465	465	6979	100
DDG 5	APR 30 75	APR 74	DEC 74	N	6514		465	6979	6979	465	465	6979	100
DDG 5	MAY 31 75	APR 74	DEC 74	N	6514		465	6979	6979	465	465	6979	100
DDG 5	JUN 30 75	APR 74	DEC 74	N	6514		465	6979	6979	465	465	6979	100
DDG 6	OCT 31 74			NONE	6450		240	6690	6596	146	240	6690	
DDG 6	NOV 30 74	APR 74	JAN 75	PH	6450		240	6690	6690	240	240	6690	
DDG 6	JAN 31 75	APR 74	JAN 75	PH	6450		260	6710	6710	260	360	6810	

MONTHLY FUNDING PROFILE

CLASS DDG SHIPS - ATLANTIC FLEET

REPORT WES WES001 DEC 29 1975

HULL	ASOF DATE	START DATE	END DATE	FCLTY	PYRYR FND	CHORD	CNTYR PLAN	CNT PLAN TOTAL	TOTFND ALLYR TODTE	CNT YEAR FNDG	EST CNTYR RQMT	EST OVHL	PERCNT OVHL
DDG 6	FEB 28 75	APR 74	FEB 75	PH	6450		260	6710	6710	260	260	6710	100
DDG 6	MAR 31 75	APR 74	FEB 75	PH	6450		260	6710	6710	260	260	6710	100
DDG 6	APR 30 75	APR 74	FEB 75	PH	6450		260	6710	6710	260	260	6710	100
DDG 6	MAY 31 75	APR 74	FEB 75	PH	6450		260	6710	6710	260	260	6710	100
DDG 6	JUN 30 75	APR 74	FEB 75	PH	6450		260	6710	6710	260	260	6710	100
DDG 6	DEC 31 75	APR 74	JAN 75	PH	6450		260	6710	6710	260	260	6710	

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS
QUERY WES
IF ASOFDATE IS BE TO 740731/750831 AND
PREFIX HULL IS DDG AND
SUFFIX HULL IS 18 OR 5 OR 6 OR 11 OR 37 OR 17 AND
SUFFIX FNDACTVTY IS COR OR CFY OR SRA OR AFR OR CORR OR CFYR OR
AFRR?
REPORT WES
SORT ON HULL AND ASOFDATE
HEAD 1 $CLASS DDG SHIPS - ATLANTIC FLEET$
$END

YOUR QUERY HAS BEEN ACCEPTED
```

```
FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY WES, QUESTION NO.   1, HAS         81 HITS-T
%A

USER ABORT

                                                    MONTHLY FUNDING PRO

                                                 CLASS DDG SHIPS - ATLAN

REPORT WES  WESØØ1

  HULL        ASOF         START        END        FCLTY   PYRYR   CHORD
              DATE         DATE         DATE                FND


DDG 11      SEP 3Ø 74   JUN 74      JUL 75      C        6534
DDG 11      OCT 31 74   JUN 74      JUL 75      C        6534
DDG

DDG 11      NOV 3Ø 74   JUN 74      JUL 75      C        6534
DDG 11      DEC %A
Ø


USER ABORT
QUERY RUN ABORTED IN ROUTINE REPGN

WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?-

PLEASE ANSWER YES OR NO-
  SIS ERROR RECOVERY
 FOLLOWING FILES TO BE CLOSED BY SIS
  RECOVERY-
          REPDEF3
YES


PLEASE INPUT QUERY COMMANDS

QUERY WES
IF ASOFDATE IS BE TO 74Ø731/75Ø831 AND
PREFIX HULL IS DDG AND
SUFFIX HULL IS 18 OR 5 OR 6 OR 11 OR 37 OR 17 AND
SUFFIX FNDACTVTY IS COR OR CFY OR SRA OR AFR OR CORR OR CFYR OR
AFRR?
REPORT WES
SORT ON HULL AND ASOFDATE
HEAD 1 $CLASS DDG SHIPS - ATLANTIC FLEET$
$END

YOUR QUERY HAS BEEN ACCEPTED
```

FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY WES, QUESTION NO.   1, HAS   81 HITS-B

MONTHLY FUNDING PROFILE

CLASS DDG SHIPS - ATLANTIC FLEET

REPORT WES WES001 DEC 29 1975

HULL	ASOF DATE	START DATE	END DATE	FCLTY	PYRYR FND	CHORD	CNTYR PLAN	CNT PLAN TOTAL	TOTFND ALLYR TODTE	CNT YEAR FNDG	EST CNTYR RQMT	EST OVHL	PERCNT OVHL
DDG 11	SEP 30 74	JUN 74	JUL 75	C	6534		8	6542	6534		108	6642	98
DDG 11	OCT 31 74	JUN 74	JUL 75	C	6534		38	6572	6542	8	38	6572	98
DDG 11	NOV 30 74	JUN 74	JUL 75	C	6534		38	6572	6542	8	38	6572	98
DDG 11	DEC 31 74	JUN 74	JUL 75	C	6534		8	6542	6542	8	8	6542	98
DDG 11	JAN 31 75	JUN 74	JUL 75	C	6534		8	6542	6542	8	258	6792	98
DDG 11	FEB 28 75	JUN 74	JUL 75	C	6534		8	6542	6542	8	258	6792	96
DDG 11	MAR 31 75	JUN 74	JUL 75	C	6534		280	6814	6542	8	280	7046	97
DDG 11	APR 30 75	JUN 74	JUL 75	C	6766		280	7046	6801	35	280	7046	100
DDG 11	MAY 31 75	JUN 74	JUL 75	C	6766		50	6816	6816	50	50	6816	100
DDG 11	JUN 30 75	JUN 74	MAY 75	C	6766		50	6816	6816	50	50	6816	100
DDG 17	JUL 31 74	OCT 74	AUG 75	N	658		7497	8155	758	100	7497	8155	
DDG 17	AUG 31 74	OCT 74	AUG 75	N	658		7497	8155	758	100	7497	8155	
DDG 17	SEP 30 74	OCT 74	AUG 75	N	658		7497	8155	1772	1114	7878	8536	96
DDG 17	OCT 31 74	OCT 74	AUG 75	N	658		7397	8055	3772	3114	8668	9326	96
DDG 17	NOV 30 74	OCT 74	SEP 75	N	658		7397	8055	7574	6916	8668	9326	96
DDG 17	DEC 31 74	OCT 74	SEP 75	N	658		7397	8055	7574	6916	8668	9326	96
DDG 17	JAN 31 75	OCT 74	SEP 75	N	658		7397	8055	7740	7082	7597	8255	96
DDG 17	FEB 28 75	OCT 74	SEP 75	N	658		7397	8055	7778	7120	7597	8255	98
DDG 17	MAR 31 75	OCT 74	SEP 75	N	658		7497	8155	7833	7175	7497	8155	100
DDG 17	APR 30 75	OCT 74	SEP 75	N	658		7497	8155	7872	7214	7497	8155	100
DDG 17	MAY 31 75	OCT 74	SEP 75	N	658		7497	8155	7894	7236	7497	8155	100
DDG 17	JUN 30 75	OCT 74	AUG 75	N	658		7410	8068	8068	7410	7410	8068	100
DDG 17	JUL 31 75	OCT 75	AUG 75		8068		20	8088	8087	19	20	8088	100
DDG 17	AUG 31 75	OCT 75	AUG 75	N	8068		20	8088	8087	19	20	8088	100
DDG 18	SEP 30 74	JAN 74	DEC 74	C	5483		219	5702	5702	219	319	5802	98
DDG 18	OCT 31 74	JAN 74	DEC 74	C	5483		254	5737	5702	219	254	5737	98
DDG 18	NOV 30 74	JAN 74	DEC 74	C	5483		254	5737	5702	219	254	5737	98
DDG 18	DEC 31 74	JAN 74	DEC 74	C	5483		254	5737	5702	219	254	5737	98
DDG 18	JAN 31 75	JAN 74	DEC 74	C	5483		197	5680	5680	197	197	5680	98
DDG 18	FEB 28 75	JAN 74	DEC 74	C	5483		197	5680	5680	197	197	5680	100
DDG 18	MAR 31 75	JAN 74	DEC 74	C	5483		197	5680	5680	197	197	5680	100
DDG 18	APR 30 75	JAN 74	DEC 74	C	5483		197	5680	5680	197	197	5680	100
DDG 18	MAY 31 75	JAN 74	DEC 74	C	5483		197	5680	5680	197	197	5680	100
DDG 18	JUN 30 75	JAN 74	DEC 74	C	5483		197	5680	5680	197	197	5680	100
DDG 2B5	DEC 31 74	JAN 75	NOV 75	PH	405		7882	8287	4425	4020	9382	9787	84
DDG 35	JUL 31 75	MAY 77	MAY 78	PH			500				500		
DDG 35	AUG 31 75	MAY 77	MAY 78	PH			300				300		
DDG 37	JUL 31 74	NOV 73	DEC 74	PH	8513			8513	8513		500	9013	
DDG 37	AUG 31 74	NOV 73	DEC 74	PH	8513			8513	8513		775	9288	
DDG 37	SEP 30 74	NOV 73	DEC 74	PH	8513		200	8713	8513	0	350	8863	98
DDG 37	OCT 31 74	NOV 73	DEC 74	PH	8982		155	9137	9117	135	155	9137	
DDG 37	NOV 30 74	NOV 73	JAN 75	PH	8982		155	9137	9137	155	155	9137	98
DDG 37	DEC 31 74	NOV 73	JAN 75	PH	8982		155	9137	9137	155	155	9137	98
DDG 37	JAN 31 75	NOV 73	JAN 75	PH	8982		325	9307	9307	325	325	9307	98
DDG 37	FEB 28 75	NOV 73	JAN 75	PH	8982		325	9307	9307	325	325	9307	100
DDG 37	MAR 31 75	NOV 73	JAN 75	PH	8982		325	9307	9307	325	325	9307	100
DDG 37	APR 30 75	NOV 73	JAN 75	PH	8982		325	9307	9307	325	325	9307	100
DDG 37	MAY 31 75	NOV 73	JAN 75	PH	8982		325	9307	9307	325	325	9307	100
DDG 37	JUN 30 75	NOV 73	JAN 75	PH	8982		325	9307	9307	325	325	9307	100

MONTHLY FUNDING PROFILE

CLASS DDG SHIPS - ATLANTIC FLEET

REPORT WES WES001 DEC 29 1975

HULL	ASOF DATE	START DATE	END DATE	FCLTY	PYRYR FND	CHORD	CNTYR PLAN	CNT PLAN TOTAL	TOTFND ALLYR TODTE	CNT YEAR FNDG	EST CNTYR RQMT	EST OVHL	PERCNT OVHL
DDG 45	JUL 31 74	APR 75	DEC 75	C	205		7205	7410	705	500	7205	7410	
DDG 45	AUG 31 74	APR 75	DEC 75	C	205		7205	7410	705	500	7205	7410	
DDG 45	SEP 30 74	APR 75	DEC 75	C	205		7205	7410	705	527	7205	7410	
DDG 45	OCT 31 74	APR 75	DEC 75	C	205		6605	6810	732	527	7205	7410	
DDG 45	NOV 30 74	APR 75	FEB 76	C	205		6605	6810	732	527	7205	7410	
DDG 45	DEC 31 74	APR 75	FEB 76	C	530		5789	6319	1007	477	6389	6919	
DDG 45	JAN 31 75	APR 75	FEB 76	C	205		7705	7910	1281	1076	7705	7910	
DDG 45	FEB 28 75	APR 75	MAR 76	C	205	500	7705	8410	3781	3076	7705	8410	100
DDG 45	MAR 31 75	APR 75	MAR 76	C	205	500	7705	8410	3781	3076	7705	8410	100
DDG 45	APR 30 75	APR 75	MAY 76		205	500	7205	7910	6931	6226	7205	7910	100
DDG 45	MAY 31 75	APR 75	MAY 76	C	205	500	7590	8295	6929	6224	7509	8295	100
DDG 45	JUN 30 75	APR 74	MAY 76	C	205		7079	7284	7284	7079	7079	8295	100
DDG 45	JUL 31 75	APR 75	MAY 76	C	7284		600	7884	7410	126	600	7884	100
DDG 45	AUG 31 75	APR 75	MAY 76	C	7284		600	7884	7410	126	600	7884	100
DDG 5	SEP 30 74	APR 74	DEC 74	N	6514		400	6914	6514		400	6914	
DDG 5	OCT 31 74	APR 74	DEC 74	N	6514		434	6948	6514	0	434	6948	
DDG 5	NOV 30 74	APR 74	DEC 74	N	6514		434	6948	6948	434	434	6948	
DDG 5	DEC 31 74	APR 74	DEC 74	N	6514		465	6979	6979	465	465	6979	
DDG 5	JAN 31 75	APR 74	DEC 74	N	6514		465	6979	6979	465	465	6979	
DDG 5	FEB 28 75	APR 74	DEC 74	N	6514		465	6979	6979	465	465	6979	100
DDG 5	MAR 31 75	APR 74	DEC 74	N	6514		465	6979	6979	465	465	6979	100
DDG 5	APR 30 75	APR 74	DEC 74	N	6514		465	6979	6979	465	465	6979	100
DDG 5	MAY 31 75	APR 74	DEC 74	N	6514		465	6979	6979	465	465	6979	100
DDG 5	JUN 30 75	APR 74	DEC 74	N	6514		465	6979	6979	465	465	6979	100
DDG 6	OCT 31 74			NONE	6450		240	6690	6596	146	240	6690	
DDG 6	NOV 30 74	APR 74	JAN 75	PH	6450		240	6690	6690	240	240	6690	
DDG 6	JAN 31 75	APR 74	JAN 75	PH	6450		260	6710	6710	260	360	6810	
DDG 6	FEB 28 75	APR 74	FEB 75	PH	6450		260	6710	6710	260	260	6710	100
DDG 6	MAR 31 75	APR 74	FEB 75	PH	6450		260	6710	6710	260	260	6710	100
DDG 6	APR 30 75	APR 74	FEB 75	PH	6450		260	6710	6710	260	260	6710	100
DDG 6	MAY 31 75	APR 74	FEB 75	PH	6450		260	6710	6710	260	260	6710	100
DDG 6	JUN 30 75	APR 74	FEB 75	PH	6450		260	6710	6710	260	260	6710	100

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- NO

???? REPDEF

 PLEASE INPUT REPORT DEFINITIONS.

ID BASE2
REPORT *1
MHEAD $AVERAGE COST OF OVERHAUL$
COL HEAD 1 = $HULL$
COL HEAD 2 = $YARD$
COL HEAD 3 = $FND/ACTVTY$
COL HEAD 4 = $BDGT/LABOR/DAYS$
COL HEAD 5 = $BDGT/LABOR/MONEY$
COL HEAD 6 = $BDGT/MATL/MONEY$
COL HEAD 7 = $BDGT/UNIT/COST$
PRINT HULL, FCLTY, FNDACTVTY, BDGTLDAYS, BDGTLMNY, BDGTMMNY,
BDGTUCOST
$END
```

```
???? IQUERY


YOU HAVE SELECTED THE INTERACTIVE QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-NO


SELECT ONE OF THE FOLLOWING DATA BASES
18 LOGISTICS DOCUMENT DATA BANK
31 YURSO DATA BANK
32 MKRS-BASE-2
?? 32

PLEASE INPUT QUERY COMMANDS
QUERY *X!
IF PREFIX HULL IS LST AND
FNDACTVTY IS 74BDGT AND
PREFIX FCLTY IS BE TO Ø/9?
REPORT *1
HEAD 1 $LST APPORTIONMENT FOR FISCAL 74$
HEAD 2 $NUMERIC YARDS ONLY$$
IF PREFIX HULL IS LST AND
FNDACTVTY IS 74BDGT AND
PREFIX FCLTY IS BE TO A/Z?
REPORT *1
SORT ON HULL
HEAD 1 $LST APPORTIONMENT FOR FISCAL 74$
HEAD 2 $ALPHABETIC YARDS ONLY$
IF PREFIX HULL IS LST AND
FNDACTVTY IS 74BDGT AND
PREFIX FCLTY IS BE TO Ø/9?
REPORT *1
HEAD 1 $LST APPORETIONMENT FOR FISCAL 74$
HEAD 2 $NUMERIC YARDS ONLY$
PRINT AVERAGE BDGTLDAYS, BDGTLMNY, BDGTMMNY, BDGTUCOST
IF PREFIX HULL IS LST AND
FNDACTVTY IS 74BDGT AND
PREFIX FCLTY IS BE TO A/Z?
REPORT *1
HEAD 1 $LST APPORTIONMENT FOR FISCAL 74$
HEAD 2 $ALPHABETIC YARDS ONLY$
SORT ON HULL
PRINT AVERAGE BDGTLDAYS, BDGTLMNY, BDGTMMNY, BDGTUCOST
```

```
IF PREFIX HULL IS LST AND
FNDACTVTY IS 75BDGT AND
PREFIX FCLTY IS BE TO Ø/9?
REPORT *1
HEAD 1 $LST APPORTIONMENT FOR FISCAL 75$
NUMERIC YARDS ONLY$
HEAD 2 $NUMERIC YARDS ONLY$
PRINT AVERAGE BDGTLDAYS, BDGTLMNY, BDGTMMNY, BDGTUCOST
IF
SORT ON HULL
IF PREFIX HULL IS LST AND
FNDACTVTY IS 75BDGT AND
PREFIX FCLTY IS BE TO A/Z?
REPORT *1
SORT ON HULL
HEAD 1 $LST APPORTIONMNENT FOR FISCAL 75$
HEAD 2 $ALPHABERTIC YARDS ONLY$
PRINT AVERAGE DBDGTLDAYS, BDGTLMNY, BDGTMMNY, BDGTUCOST
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY *X!, QUESTION NO.   1, HAS          1 HITS-D

QUERY *X!, QUESTION NO.   2, HAS          Ø HITS
QUERY *X!, QUESTION NO.   3, HAS          1 HITS-T

QUERY *X!, QUESTION NO.   4, HAS          Ø HITS
QUERY *X!, QUESTION NO.   5, HAS          1 HITS-T

QUERY *X!, QUESTION NO.   6, HAS          Ø HITS
```

AVERAGE COST OF OVERHAUL

LST APPORETIONMENT FOR FISCAL 74

NUMERIC YARDS ONLY

REPORT *1 *X!ØØ3 DEC 29 1975

HULL	YARD	FND ACTVTY	BDGT LABOR DAYS	BDGT LABOR MONEY	BDGT MATL MONEY	BDGT UNIT COST
LST 1179	5	74BDGT		2895	737	3632
		****AVERAGE***				
			*	2895	737	3632

```
NSRDC 6700 INTERCOM V4.2
DATE  12/29/75
TIME  16.26.21.

PLEASE LOGIN
LOGIN,PUWEWEAVER,1180600823,SUP

WAITING FOR INPUTYES

PLEASE INPUT QUERY COMMANDS
QUERY *X!
IF PREFIX HULL IS LST AND
FNDACTVTY IS 74BDGT AND
PREFIX FCLTY IS BE TO 0/9?
REPORT *1
SORT ON HULL
PRINT AVERAGE BDGTLDAYS, BDGTLMNY, BDGTMMNY, BDGTUCOST
HEAD 1 $LIST APPORTIONMENT FOR FISCAL 74$
HEAD 2 $NUMERIC YARDS ONLY$
IF PREFIX HULL IS LST AND
FNDACTVTY IS 74BDGT AND
PREFIX FCLTY IS BE TO A/Z?
REPORT *1
SORT ON HULL
PRINT AVERAGE BDGTLDAYS, BDGTLMNY, BDGTMMNY, BDGTUCOST
HEAD 1 $LST APPORTIONMENT FOR FISCAL 74$
HEAD 2 $ALPHABETIC YARDS ONLY$$
IF PREFIX HU;LL IS LST AND
FNDACTVTY IS 75BDGT AND
PREFIX FCLTY IS BE TO 0/9 ?
REPORT *1
SORT ON HULL
PRINT AVERAGE BDGTLDAYS, BDGTLMNY, BDGTMMNY, BDGTUCOST
HEAD 1 $LST APPORTIONMENT - FISCAL 75$
HEAD 2 $NUMERIC YARDS ONLY$
IF PREFIX HULL IS LST AND
FNDACTVTY IS 75BDGT AND
PREFIX FCLTY IS BE TO A/Z?
REPORT *1
SORT ON HULL
PRINT AVERAGE BDGTLDAYS, BDGTLMNY, BDGTMMNY, BDGTUCOSAT
PRINT AVERAGE BDGTLDAYS, BDGTLMNY, BDGTMMNY, BDGTUCOST
HEAD 1 $LST APPORTIONMENT FOR FISCAL 75$
HEAD 2 $APLHABETIC YARDS ONLY$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY *X!, QUESTION NO.   1, HAS          1 HITS-T

QUERY *X!, QUESTION NO.   2, HAS          0 HITS
QUERY *X!, QUESTION NO.   3, HAS          0 HITS
```

AVERAGE COST OF OVERHAUL

LIST APPORTIONMENT FOR FISCAL 74

NUMERIC YARDS ONLY

REPORT *1 *X!001 DEC 29 1975

HULL	YARD	FND ACTVTY	BDGT LABOR DAYS	BDGT LABOR MONEY	BDGT MATL MONEY	BDGT UNIT COST
LST 1179	5	74BDGT		2895	737	3632
		****AVERAGE***				
			*	2895	737	3632

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES


PLEASE INPUT QUERY COMMANDS
QUERY *X1

IF PREFIX HULL IS DE OR FF AND
FNDACTVTY IS 74BDGT AND
PREFIX FCLTY IS BE TO 0/9?
REPORT *1
SORT ON HULL
PRINT AVERAGE BDGTLDAYS, BDGTLMNY, BDGTMMNY, BDGTUCOST
HEAD 1 $DE AND FF APPORTIONMENT FOR FISCAL 74$
HEAD 2 $NUMERIC YARDS ONLY$
IF PREFIX HULL IS DE OR FF AND
FNDACTVTY IS 75BDGT AND
PREFIX FCLTY IS BE TO A/Z?
REPORT *1
SORT ON HULL
BBBBBBBBBBB
PLEASE LOGIN
LOGIN,PUWEWEAVER,1180600823,SUP

WAITING FOR INPUT
YES

 **,WE WILL BE GOING DOWN ABOUT 1900



YES
```

```
NSRDC 6700 INTERCOM V4.2
DATE  12/29/75
TIME  16.56.38.

PLEASE LOGIN
LOGIN,PUWEWEAVER,1180600823,SUP

WAITING FOR INPUT%A


USER ABORT
QUERY RUN ABORTED IN ROUTINE LT

WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?-
  SIS ERROR RECOVERY
 FOLLOWING FILES TO BE CLOSED BY SIS
  RECOVERY-
         REPDEF3
YES


PLEASE INPUT QUERY COMMANDS

QUERY *X1
IF PREFIX HULL IS DE OR FF AND
FNDACTVTY IS 74BDGT AND
PREFIX FCLTY IS BE TO 0/9?
REPORT *1
SORT ON HULL
PRINT AVERAGE BDGTLDAYS, BDGTLMNY, BDGTMMNY, BDGTUCOST
HEAD 1 $DE AND FF APPORTIONMENT FOR FISCAL 75$
HEAD 2 $NUMERIC YARDS ONLY$
IF PREFIX HULL IS DE OR FF AND
FNDACTVTY IS 75BDGT AND
PREFIX FCLTY IS BE TO A/Z?
REPORT *1
SORT O N
SORT ON HULL
PRINT AVERAGE BDGTLDAYS, BDGTLMNY, BDGTMMNY, BDGTUCOST
HEAD 1 $DE AND FF APPORTIONMENT FOR FISCAL 75$
HEAD 2 $ALPHABETIC YARDS ONLY$
$END

YOUR QUERY HAS BEEN ACCEPTED



FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY *X1, QUESTION NO.   1, HAS          0 HITS
QUERY *X1, QUESTION NO.   2, HAS          6 HITS-
T
```

AVERAGE COST OF OVERHAUL

DE AND FF APPORTIONMENT FOR FISCAL 75

ALPHABETIC YARDS ONLY

REPORT *1 *X1002 DEC 29 1975

HULL	YARD	FND ACTVTY	BDGT LABOR DAYS	BDGT LABOR MONEY	BDGT MATL MONEY	BDGT UNIT COST
FF 1038	PH	75BDGT	18346	2189	389	2578
FF 1047	C	75BDGT	38426	4954	761	5715
FF 1059	PH	75BDGT	38322	4572	754	5325
FF 1072	C	75BDGT	37602	4848	760	5608
FFG 5	PH	75BDGT	47576	5676	829	6505
FFG 6	C	75BDGT	48250	6221	837	7038
		****AVERAGE***				
			38087	4743	722	5462

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS

QUERY *X2
IF PREFIX HULL IS DE OR FF AND
FNDACTVTY IS 74BDGT AND
PREFIX FCLTY IS BE TO A/Z?
REPORT *1
SORT ON HULL
PRINT AVERAGE BDGTLDAYS, BDGTLMNY, BDGTMNY
PRINT AVERAGE BDGTLDAYS, BDGTLMNY, BDGTMMNY, BDGTUCOST
HEAD 1 $DE AND FF APPORTIONMENT FOR FISCAL 74$
HEAD 2 $ALPHABETIC YARDS ONLY$
IF PREFIX HULL IS DE OR FF AND
FNDACTVTY IS 75BDGT AND
PREFIX FCLTY IS BE TO Ø/9?
REPORT *1
SORT ON HULL
HEAD 1 $DE AND FF APPORTIONMENT FOR FIDCAL 75$
HEAD 2 $NUMERIC YARDS ONLY$
PRINT AVERAGE,
PRINT AVERAGE BDGTLDAYS, BDGTLMNY, BDGTMMNY, BDGTUCOST
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY *X2, QUESTION NO.   1, HAS          2 HITS-T

QUERY *X2, QUESTION NO.   2, HAS          Ø HITS T
```

AVERAGE COST OF OVERHAUL

DE AND FF APPORTIONMENT FOR FISCAL 74

ALPHABETIC YARDS ONLY

REPORT *1 *X2001 DEC 29 1975

HULL	YARD	FND ACTVTY	BDGT LABOR DAYS	BDGT LABOR MONEY	BDGT MATL MONEY	BDGT UNIT COST
FF 1049	C	74BDGT	21319	2601	542	3143
FF 1056	PH	74BDGT	21051	2337	422	2759
		****AVERAGE***				
			21185	2469	482	2951

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?-

PLEASE ANSWER YES OR NO- YES


PLEASE INPUT QUERY COMMANDS
QUERY *X3
IF ASOFDATE IS 750630 AND ENDATE IS LE TO
IF ASOFDATE IS 750630 AND PREFIX ENDATE IS LE TO 7506 AND
PREFIX HULL IS DE OR FF AND
PREFIX FCLTY IS BE TO 0/9?
REPORT *2
SORT ON HULL
PRINT AVERAGE TOTFND
HEAD 1 $DE AND FF CLASS SHIPS$
HEAD 2 $NUMERIC YARDS ONLY$
IF ASOFDATE IS 750630 AND ENDATE IS LE TO 7506 AND
PREFIX HULL IS LST AND
PREFIX FCLTY IS BE TO 0/9?
REPORT *2
SORT ON HULL
PRINT AVERAGE TOTFND
HEAD 1 $LST CLASS SHIPS$
HEAD 2 $AVERAGE OVERHAUL COST FOR FISCAL 75$
HEAD 3 $NUMERIC YARDS ONLY$
IF  ASOFDATE IS 750630 AND ENDATE IS LE
%A


USER ABORT
QUERY RUN ABORTED IN ROUTINE LT

WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?-
  SIS ERROR RECOVERY
 FOLLOWING FILES TO BE CLOSED BY SIS
  RECOVERY-
           REPDEF3
YES
```

```
PLEASE INPUT QUERY COMMANDS
QUERY *X3
IF ASOFDATE IS 750630 AND PREFIX ENDATE IS LE TO 7506 AND
PREFIX HULL IS DE OR FF AND
PREFIX FCLTY IS BE TO 0/9?
REPORT *2
SORT ON HULL
PRINT AVERAGE TOTFND
HEAD 1 $DE AND FF CLASS SHIPS$
HEAD 2 $AVERAGE OVERHAUL COST FOR FISCAL 75$
HEAD 3 $NUMERIC YARDS ONLY$
IF ASOFDATE IS 750630 AND PREFIX ENDATE IS LE TO 7506 AND
PREFIX HULL IS DE OR FF AND
PREFIX FCLTY IS BE TO A/Z?
REPORT *2
SORT ON HULL
PRINT AVERAGE TOTFND
HEAD 1 $DE AND FF CLASS SHIPS$
HEAD 2 $AVERAGE OVERHAUL COST FOR FISCAL 75$
HEAD 3 $APLPHABETIC YARDS ONLY$
IF ASOFDATE IS 750630 AND PREFIX ENDATE IS LE TO 7506 AND
PREFIX HULL IS LST AND
PREFIX FCLTY IS BE TO 0/9?
REPORT *2
SORT ON HULL
PRINT ASVERAGE
PRINT AVERAGE TOTFND
HEAD 1 $LST CLASS SHIP$
HEAD 2 $AVERAGE OVERHAUL COST FOR FISCAL 75$
HEAD 3 $NUMERIC YARDS ONLY$
IF ASOFDATE IS 750630 AND PREFIX ENDATE IS LE TO 7506 AND
PREFIX HULL IS LST AND
PREFIX FCLTY ISBE TO A/Z?
REPORT *2
SORT ON HUL;L
SORT ON HULL
PRINT AVERAGE TOTFND
HEAD 1 $LST CLASS SHIPS$
HEAD 2 $AVERAGE OVERHAUL COST FOR FISCAL 75$
HEAD 3 $ALPHABETIC YARDS ONLY$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY *X3, QUESTION NO.   1, HAS          0 HITS
QUERY *X3, QUESTION NO.   2, HAS          1 HITS-T

QUERY *X3, QUESTION NO.   3, HAS          0 HITS
QUERY *X3, QUESTION NO.   4, HAS          0 HITS T
```

AVERAGE COST OF OVERHAUL

DE AND FF CLASS SHIPS

AVERAGE OVERHAUL COST FOR FISCAL 75

ALPHABETIC YARDS ONLY

REPORT *2 *X3002 DEC 29 1975

HULL	YARD	FND ACTVTY	END DATE	TOTAL FNDG	PERCNT OVHL
FF 1049	C	75COR	APR 75	4949	100

****AVERAGE***

4949

```
NSRDC 6700 INTERCOM V4.2
DATE  12/30/75
TIME  14.08.41.

PLEASE LOGIN
LOGIN,PUWEWEAVER,1180600823,SUP

COMMAND- ETL,500

COMMAND- COMRADE,SHARP


COMRADE        TIME:       14.09.38.
               DATE:       12/30/75

SHARP SUBSYSTEM ENTERED
???? IQUERY


YOU HAVE SELECTED THE INTERACTIVE QUERY OPTION.

WOULD YOU LIKE AN EXPLANATION OF THE SYSTEM?-NO


SELECT ONE OF THE FOLLOWING DATA BASES
18 LOGISTICS DOCUMENT DATA BANK
31 YURSO DATA BANK
32 MKRS-BASE-2
?? 32


PLEASE INPUT QUERY COMMANDS
QUERY QT
IF PREFIX HULL IS LST AND
FNDACTVTY IS 75BDGT AND
PREFIX FCLTY IS BE TO 0/9?
REPORT *1
SORT ON HULL
PRINT AVERAGE BDGTLDAYS, BDGTLMNY, BDGTMMNY, BGDGTUCOST
HEAD 1 $LST APPORTIONMENT - FISCAL 75$
HEAD 2 $NUMERIC YARDS ONLY$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY QT , QUESTION NO.   1, HAS          1 HITS-T
```

AVERAGE COST OF OVERHAUL

LST APPORTIONMENT - FISCAL 75

NUMERIC YARDS ONLY

REPORT *1 QT 001 DEC 30 1975

HULL	YARD	FND ACTVTY	BDGT LABOR DAYS	BDGT LABOR MONEY	BDGT MATL MONEY	BDGT UNIT COST
LST 1170	5	75BDGT		3292	778	4070
		****AVERAGE***				
			*	3292	778	4070

```
WOULD YOU LIKE TO SUBMIT ANOTHER QUERY?- YES

PLEASE INPUT QUERY COMMANDS
QUERY TWX
IF PREFIX RCDID IS 411?
HEAD 1 $SHARP QUERY$
HEAD 2 $INTEGRATION OF THE DATA BASE/COMMUNICATIONS FUNCTION$
HEAD 3 $FLEXIBILITY OF THE SPLIT-PHASE KEY$
$END

YOUR QUERY HAS BEEN ACCEPTED


FOR EACH QUESTION, TYPE IN D, T, OR B.

QUERY TWX, QUESTION NO.   1, HAS          2 HITS-T
```

SHARP QUERY

INTEGRATION OF THE DATA BASE/COMMUNICATIONS FUNCTION

FLEXIBILITY OF THE SPLIT-PHASE KEY

REPORT DFT TWX001 DEC 30 1975

PROC/DATE: 750702
RCDID: 411010001
TWX: SHARP WILL YOU CHECK UPDATE?

PROC/DATE: 750710
RCDID: 411020002
TWX: WEAVER/SHARP REFERENCE YOUR MESSAGE 411010001. UPDATE IS OPERATIONAL. YOUR TWX HAS BEEN RECEIVED HERE AT NSRDC. NEW SUBJECT: REGARDING THE QUERY TO RETRIEVE SUBMARIMES, I MAY HAVE GIVEN CHUCK THE WRONG ACRONYM TO BE USED FOR THE NC VALUE. IF I GAVE HIM OVHLACTVTY IN WRITING ON THE PRINTOUT, IT WAS WRONG. INSTEAD OF USING FIELD 35 I SUGGEST YOU USE FIELD 37 WHICH IS TYPEOVHL I.E. AND TYPEOVHL IS NC? NO HITS ON THAT ONE BUT IF YOU TRY ONE THAT YOU KNOW IS IN THERE IT SHOULD WORK. I HAVE OBTAINE CORRECT RESULTS ON THAT QUERY HERE AT THE NSRDC SITE. EOM

INITIAL DISTRIBUTION

Copies		
1	ARPA/INFO PROC TECH	
3	ASN	
	1 FM	
	1 R&D	
	1 I&L	
6	CNO	
	1 OP 91	
	2 OP 917 (Mr. Braxton)	
	1 OP 40	
	2 OP 401G (Mr. Gruber)	
1	CHONR (Mr. Dennicoff)	
2	NAVMAT	
	1 MAT 04 (Mr. McGuire)	
	1 MAT 03 (CDR Dieterle)	
1	CINCLANTFLT (CDR Sharp)	
1	CINCPACFLT (Mr. Chase)	
1	NELC (NAICOMMIS DEV. GROUP)	
2	DPSCPAC (CDR Buckner)	
2	DPSCLANT (Mr. Palacio)	
1	NMCSA (Mr. Ellis)	
1	NAVSEA 934(H) (Mr. Welsh)	
1	NAVSUP 04	
1	FMSO 96	
2	NAVCOSSACT (Mr. Rogers) PERA (CRUDES) Phila. (Mr. Morrison)	
12	DDC	
1	General Electric Company (MATSCO) (Mr. Weaver)	
1	Rockwell International (Mr. Tipton)	

CENTER DISTRIBUTION

Copies	Code
1	18
1	1804
1	1805
20	1806
1	1809
1	183
1	184
1	185
1	186
1	188
4	1882
1	189
30	5214.1
1	5221
1	5222

PRECEDING PAGE BLANK-NOT FILMED